TM03/TE16

CONTROL LOGIC TEST 1

CZTEAB0

AH-A792B-MC

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA

```
.REM    %
```

IDENTIFICATION

PRODUCT CODE:	AC-A791B-MC
PRODUCT NAME:	CZTEAB0 TM03-TE16/TU77 CONTROL LOGIC TEST PART I
DATE CREATED:	15 NOV 78
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	J. G. ADAMS

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. TE16 MAGTAPE TRANSPORT

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE.

4. STARTING PROCEDURE

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

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

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

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

4.1 AUTOMATIC MODE OPERATION

IF THIS PROGRAM IS LOADED & RUN UNDER AUTOMATIC (CHAIN) MODES
DEFAULT RESPONSES TO OPERATOR REQUESTS ARE USED, AND THE SOFTWARE
SWR INVOKED WITH A SWITCH SETTING OF 000000 . NO
OPERATOR INTERVENTION IS REQUIRED. IN ORDER TO SET THE SWR TO
A DIFFERENT SETTING, CHANGE LOC:176(SWREG) TO THE DESIRED SETTING.

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

**NOTE:THIS PROGRAM IS DEFAULTED TO EXPECT SLAVE TYPE
AS TE16; IF IT IS TO BE RUN IN CHAIN MODE ON A TU77
THE SLAVE TYPE WORD (SLVTYP: AT LOC:1010) MUST BE SET
TO A ONE INSTEAD OF A ZERO. (TE16=0/TU77=1)

**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).

** NON-STANDARD JUMPER MODE
M8931 (W2 IN) ,M8937 (W2 IN,W1 OUT) **

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

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

REGISTER START: <172440> (CR)                   [REGS:]
VECTOR ADDRESS: <224> (CR)                      [VECT:]
IS CONTROLLER JUMPERED IN NON-STANDARD MODE
TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD <3>  [JUMPER:]
TM03 DRIVE: <0> (CR)                            [DRVN:]
TE16 SLAVE: <0> (CR)                            [SLVN:]
STATIC TESTS ONLY: <0> (CR)                     [STATC:]
SLAVE TYPE (0=TE16,1 TU77): <0> (CR)            [SLVTYP:]
        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 (M8909 OR RH)
NON-EXIST DRIVE 3 EXPT-NOT RECVD
ITER: 3
```

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

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

```
LOGIC TEST 7: FC BIT TEST (M8705)
FC BITS 15-0 EXPT 177777 RECVD 177577
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7. OPERATION

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

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

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

SINGLE TEST SELECTION: (SW0-SW5)

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

8. SUB TEST SUMMARIES

LOGIC TEST #1: DRIVE ADDRESSING

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

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

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8909

CIRCUITS	PRINT REFERENCES
RH-DS BITS	(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,M8909,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,M8909,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 TE16 AND THE SPR BIT.

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

PROGRAMMED SEQUENCE: THE SLAVE ADDRFSS 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,M8910,M9001YA,M8933

CIRCUITS	PRINT REFERENCE
REGISTER SELECT	(MBI2)
SLAVE ADDRESS BITS	(MR6)
SLAVE ADDRESS LINES	(M8937,2-2),(LAW6)
TE16 ADDRESS DECODE	(LAW6)
SPR BIT	(LAW6)(M9001YA)(TCCM7)

LOGIC TEST #5: MAINTENANCE REGISTER BITS

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

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

LIKELY FAULT LOCATIONS: M8905-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: M8909,M8905-YB

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

LOGIC TEST #7: FRAME COUNT BIT TEST

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

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

LIKELY FAULT LOCATIONS: M8909

CIRCUITS	PRINT REFERENCE
TM03 REGISTER SELECT	(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: M8909, 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),M8909,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: M8910,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: M8910,M8933,M8909,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: M8910,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: M8909

CIRCUIT	PRINT REFERENCE
SET ILF DECODE	MBI5,MBI7
ILF F_OP	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: M8909

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: M8909

CIRCUIT	PRINT REFERENCE
MASSBUS PARITY TREE	MBI4
CPAR FLOP	MBI[illegible]
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,M8909

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: M8909

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: M8909, MB CABLE, M8933, M8905-YB

CIRCUITS	PRINT REFERENCE
RUN LINE	MB*
EBL PLS	MBI9
FCE FLOP	MBI1*
SHUTDOWN LOGIC	TCCM5
MAINT. FUNCTION DECODE	MR5

LOGIC TEST #27: ILLEGAL REGISTER

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

LIKELY FAULT LOCATIONS: MASSBUSS, M8909

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

LOGIC TEST #30: DRIVE TIMING ERROR

PROGRAMMED SEQUENCE:

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

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

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

LOGIC TEST 31: OPERATION INCOMPLETE (OPI)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8933, M8909

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. IF THE DRIVE TYPE REG INDICATES A TU77
THEN NON-EXECUTABLE FUNCTION (NEF) IS ALSO CHECKED.

LIKELY FAULT LOCATIONS: M8909, M8910, 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: M8909, M8933

CIRCUITS	PRINT REFERENCES
SPACE FUNCTION DECODE	M8I5
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, M8931, 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: M8909, 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, M8931

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

LOGIC TEST 42: CYCLIC REDUNDANCY ERROR

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905-YB, M8934, G056, SLAVE CABLE, M8910

CIRCUITS	PRINT REFERENCES
MM FUNCTION DECODE	MR5
CRC CHECK CIRCUIT	CNRZ3

LOGIC TEST 43: LRC

PROGRAMMED SEQUENCE:

A WRITE OPERATION IS PERFORMED WITH A MM FUNCTION (INC TMRL) WHICH ASSERTS WD(SB) 5L THROUGHOUT THE RECORD. ALL ONES DATA IS USED SO THAT THE FUNCTION ONLY INTERFERES WITH THE WRITING OF THE LRC CHARACTER WHEN NONE OF THE TM03 WRITE DATA LINES SHOULD BE ASSERTED.
** NOTE: THIS TEST IS NOT PERFORMED ON A TU77 SLAVE.

LIKELY FAULT LOCATIONS: M8505, M8933, M8910, 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 CORRECTABLE 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: M8909

FRAME COUNT REGISTER MBI8

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

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

PROGRAM SEQUENCE:

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

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

THIS TEST IS THE COMPLEMENT OF LOGIC TEST 50 ABOVE.

%

```
1185                                    .LIST   BIN,LOC,SEQ
1186                                    .TITLE CZTEAB0 TM03-TE16/TU77 CTL I
1187                                    ;CONTROL LOGIC TEST PART I
1188                                    ;AC-A791B-MC
1189                                    ;FEB 77
1190                                    ;J.G. ADAMS
1191                                    ;REVISED MAY 1978 BY J. G. ADAMS ;++B CHANGED MODULE REFERENCES TO
1192                                                                     ;++B REFLECT TM03 MODULES
1193                                                                    ;++B ADDED TU77 TEST CAPABILITY
1194                                    ;REVISED NOV 1978 BY M. PAGE     ;+ INDICATES ENHANCEMENTS TO
1195                                    ;                                   THE ORIGINAL REV (DZTEAA)
1196                                    ;                                   NECESSARY FOR TM03
1197                                :                                  NON-STANDARD JUMPER MODE (SEE 4.2 IN DOC.)
1198                                    .MCALL  .$ACT11,.$EOP,$CATCH,$SAVE,$RESTORE,$CHAIN,$CHNMODE
1199                                    .NLIST  MC
1200                                    .LIST   ME
1201                                    .ENABLE ABS,AMA
1202
1203
1204                                    ;CONSOLE SWITCHES*********
1205                                    ;
1206                                    ;SW15:  1=HALT ON ERROR
1207                                    ;       0=CONTINUE
1208                                    ;SW14:  1=LOOP ON ERROR
1209                                    ;       0=CONTINUE
1210                                    ;SW13:  1=DO NOT PRINT ERRORS
1211                                    ;       0=PRINT ERRORS
1212                                    ;SW12:  0=HALT AT END OF PASS
1213                                    ;       1=CONTINUOUS CYCLE
1214                                    ;SW11:  1=INHIBIT ITERATIONS
1215                                    ;       0=DO ITERATIONS
1216                                    ;SW10:  1=HALT AT END OF EACH TEST
1217                                    ;       0=CONTINUE
1218                                    ;SW9:   1=DO MANUAL INTERVENTION TESTS
1219                                    ;       0=INHIBIT MANUAL INTERVENTION
1220                                    ;SW0-5: SELECT TEST NUMBER :: 00 ALL TESTS
```

```
                        ;TE16 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;FRR;PIP;MOL;WRL;EOT;XXX;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
;ATN:                   ; 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
```

```
1267                                    ;REGISTER EQUIVS**********
1268
1269            000000                  R0=%0
1270            000001                  R1=%1
1271            000002                  R2=%2
1272            000003                  R3=%3
1273            000004                  R4=%4
1274            000005                  R5=%5
1275            000006                  SP=%6
1276            000007                  PC=%7
1277
1278
1279                                ;ACT11 HOOK ***********
1280            000764                  $SVPC=.                 ;SAVE CURRENT LOCATION CTR
1281            000046                  .-46
1282  000046    002616                  .WORD   $ENDAD          ;SET LOCATION 46
1283            000052                  .=52
1284  000052    000000                  .WORD   0               ;SET LOCATION 52 = 0
1285            000764                  .-$SVPC                 ;RESTORE LOCATION CTR
1286
1287                                    ;TTY INTERRUPT VECTOR**********
1288
1289            000060                  .-60
1290  000060    017212                  .WORD   TTINT   ;TTY INTERRUPT HEADER ADDRESS
1291  000062    000340                  .WORD   340     ;PRIORITY LEVEL 7
1292
1293                                ;SOFTWARE SWITCH REGISTER***********
1294                                ;USED IF HARDWARE SWR = 177777 OR NOT AVAILABLE
1295            000176                  .=176
1296  000176    000000            SWREG:.WORD   0                       ;SOFTWARE SWITCH REGISTER
1297
1298                                    ;START ADDRESS**********
1299            000200                  .=200
1300  000200    000137  001332            JMP     START   ;PROGRAM START
1301
1302                                ;RESTART ADDRESS**********
1303            000210                  .-210
1304  000210    000137  002202            JMP     ST2
1305
1306                                    ;TM03 INTERRUPT VECTOR**********
1307
1308            000224                  .=224
1309  000224    017202                  MTINT           ;TAPE INTERRUPT HANDLER ADDRESS
1310  000226    000340                  340
1311
```

```
1312
1313            000510                          .=510
1314                                            ;MASS BUS REGISTER EQUIVS**********
1315
1316  000510    172440            C1:           172440
1317  000512    172442            WC:           172442
1318  000514    172444            BA:           172444
1319  000516    172446            FC:           172446
1320  000520    172450            CS:           172450
1321  000522    172452            DS:           172452
1322  000524    172454            ER:           172454
1323  000526    172456            AS:           172456
1324  000530    172460            CC:           172460
1325  000532    172462            DB:           172462
1326  000534    172464            MR:           172464
1327  000536    172466            DT:           172466
1328  000540    172470            SN:           172470
1329  000542    172472            TC:           172472
1330
1331                                            ;ILLEGAL FUNCTION CODES
1332
1333  000544    005405            ILFT:         5405
1334  000546    007415                          7415
1335  000550    016423                          16423
1336  000552    020437                          20437
1337  000554    022443                          22443
1338  000556    025447                          25447
1339  000560    031455                          31455
1340  000562    033465                          33465
1341  000564    036473                          36473
1342
1343                                            ;CONSTANTS**********
1344
1345  000566    177776            PSW:          177776          ;PROCESSOR STATUS
1346  000570    177570            SWR:          177570          ;SWITCH REGISTER
1347  000572    177560            TKS:          177560          ;TTY READER STATUS
1348  000574    177562            TKB:          177562          ;TTY READ BUFFER
1349  000576    177564            TPS:          177564          ;TTY PUNCH STATUS
1350  000600    177566            TPB:          177566          ;TTY PUNCH BUFFER
1351  000602    000020            ITAMT:        20              ;ITERATION AMOUNT
1352  000604    000224            VECT:         224             ;INTERRUPT VECTOR(RH)
1353  000606    172440            REGS:         172440          ;STARTING REGISTER ADDRESS
```

```
 1354                                        ;FLAGS AND COUNTERS**********
 1355
 1356  000610  000000                TOB:    0
 1357  000612  000000                TIB:    0
 1358  000614  000000                HDRFL:  0
 1359  000616  000000                EMADDR: 0
 1360  000620  000000                DRVN:   0
 1361  000622  000000                TR00:   0
 1362  000624  000000                TR01:   0
 1363  000626  000000                TR02:   0
 1364  000630  000000                TR03:   0
 1365  000632  000000                TR04:   0
 1366  000634  000000                TR05:   0
 1367  000636  000000                TR06:   0
 1368  000640  000000                TR07:   0
 1369  000642  000000                TR10:   0
 1370  000644  000000                 R11:   0
 1371  000646  000000                TR12:   0
 1372  000650  000000                TR13:   0
 1373  000652  000000                TR14:   0
 1374  000654  000000                TR15:   0
 1375  000656  000000                NRZOF:  0
 1376  000660  000000                SLVN:   0
 1377  000662  000000                PFLG:   0
 1378  000664  000000                RTRN:   0
 1379  000666  000000                ERADD:  0
 1380  000670  000000                TEMP1:  0
 1381  000672  000000                TEMP2:  0
 1382  000674  000000                TEMP3:  0
 1383  000676  000000                ITCNT:  0
 1384  000700  000000                SAV1:   0
 1385  000702  000000                SAV2:   0
 1386  000704  000000                SAV3:   0
 1387  000706  000000                SCOLP:  0
 1388  000710  000000                ITRLP:  0
 1389  000712  000000                EXFL:   0
 1390  000714  000000                ATAF:   0
 1391  000716  000000                SLAF:   0
 1392  000720  000000                SSCF:   0
 1393  000722  000000                ERRF:   0
 1394  000724  000000                ASF:    0
 1395  000726  000000                SCF:    0
 1396  000730  000000                TREF:   0
 1397  000732  000000                PEXFL:  0
 1398  000734  000000                STFLG:  0
 1399  000736  000000                LTADD:  0
 1400  000740  000000                T24FL:  0
 1401  000742  000000                ADDFL:  0
 1402  000744  000000                WAM:    0
 1403  000746  000000                FUN:    0
 1404  000750  000000                DATC:   0
 1405  000752  000000                WTAD:   0
 1406  000754  000000                DATAD:  0
 1407  000756  000000                RDAD:   0
 1408  000760  000000                W2FLG:  0
 1409  000762  000000                DERFL:  0
```

```
1410  000764  000000                PREFL:  0
1411  000766  000000                SERFL:  0
1412  000770  000000                CRCNT:  0
1413  000772  000000                UDES:   0
1414  000774  000000                WPGFL:  0
1415  000776  000000                PATRN:  0
1416  001000  000000                STATF:  0
1417  001002  000000                RDRVF:  0
1418  001004  000000                RCDP:   0
1419  001006  000000                STATC:  0
1420  001010  000000                SLVTYP: .WORD   0               ;++B INDICATES SLAVE TYPE (0/1 - TE16/TU77)
1421  001012  000000                SKAT:   0
1422  001014  000000                PCNTR:  0               ;PASS COUNTER
1423  001016  000003                JUMPER: 3               ;+INDICATOR FOR NON-STANDARD CONFIG.
1424  001020  000000                NONSTD: 0               ;+FLAG FOR NON-STANDARD CONFIG.
1425
1426                                        ;EXPT WRAP STATUS*******
1427
1428  001022  000000                WCS1:   0
1429  001024  000000                WCS2:   0
1430  001026  000000                WDS:    0
1431  001030  000000                WER:    0
1432
1433                                        ;CORE DUMP PATTERNS*******
1434
1435  001032  000005                WCDP2:  5
1436  001034  000005                        5
1437  001036  000012                        12
1438  001040  000012                        12
1439  001042  000000                        0
1440  001044  000017                WCDP0:  17
1441  001046  000017                        17
1442  001050  000017                        17
1443  001052  000017                        17
1444  001054  000000                        0
```

```
1445
1446                                        ;LOGIC TEST ENTRY TABLE**********
1447
1448  001056  000000                TSTTBL: 0
1449  001060  000000                        0
1450  001062  002666                        LT1
1451  001064  002666                        LT1
1452  001066  003142                        LT2
1453  001070  003142                        LT2
1454  001072  003346                        LT3
1455  001074  003350                        LT3IT
1456  001076  003530                        LT4
1457  001100  003530                        LT4
1458  001102  004132                        LT5
1459  001104  004140                        LT5IT
1460  001106  004322                        LT6
1461  001110  004324                        LT6IT
1462  001112  004436                        LT7
1463  001114  004440                        LT7IT
1464  001116  004552                        LT10
1465  001120  004554                        LT10IT
1466  001122  004676                        LT11
1467  001124  004700                        LT11IT
1468  001126  005122                        LT12
1469  001130  005124                        LT12IT
1470  001132  005316                        LT13
1471  001134  005326                        LT13IT
1472  001136  005420                        LT14
1473  001140  005460                        LT14IT
1474  001142  005530                        LT15
1475  001144  005570                        LT15IT
1476  001146  005640                        LT16
1477  001150  005700                        LT16IT
1478  001152  005754                        LT17
1479  001154  006014                        LT17IT
1480  001156  006070                        LT20
1481  001160  006104                        LT20IT
1482  001162  006232                        LT21
1483  001164  006246                        LT21IT
1484  001166  006376                        LT22
1485  001170  006412                        LT22IT
1486  001172  006522                        LT23
1487  001174  006536                        LT23IT
1488  001176  006652                        LT24
1489  001200  006666                        LT24IT
1490  001202  007176                        LT25
1491  001204  007204                        LT25IT
1492  001206  007330                        LT26
1493  001210  007336                        LT26IT
1494  001212  007544                        LT27
1495  001214  007570                        LT27IT
1496  001216  007662                        LT30
1497  001220  007704                        LT30IT
1498  001222  010200                        LT31
1499  001224  010206                        LT31IT
1500  001226  011030                        LT32
```

```
1501  001230  011044                        LT32IT
1502  001232  011206                        LT33
1503  001234  011222                        LT33IT
1504  001236  011310                        LT34
1505  001240  011324                        LT34IT
1506  001242  011444                        LT35
1507  001244  011460                        LT35IT
1508  001246  011616                        LT36
1509  001250  011632                        LT36IT
1510  001252  011736                        LT37
1511  001254  011752                        LT37IT
1512  001256  012106                        LT40
1513  001260  012122                        LT40IT
1514  001262  012226                        LT41
1515  001264  012242                        LT41IT
1516  001266  012470                        LT42
1517  001270  012526                        LT42IT
1518  001272  013020                        LT43
1519  001274  013056                        LT43IT
1520  001276  013264                        LT44
1521  001300  013312                        LT44IT
1522  001302  013532                        LT45
1523  001304  013560                        LT45IT
1524  001306  013776                        LT46
1525  001310  014024                        LT46IT
1526  001312  014240                        LT47
1527  001314  014254                        LT47IT
1528  001316  014430                        LT50
1529  001320  014444                        LT50IT
1530  001322  014604                        LT51
1531  001324  014620                        LT51IT
1532  001326  002552          TADX:   .WORD   TEND
1533  001330  000051          TLAST:  .WORD   51          ;CONTAINS # OF TESTS
```

```
1534                                            .EVEN
1535                                            ;PROGRAM START AND HOUSEKEEPING**********
1536
1537                                    ;NOTE: PROGRAM STARTS HERE ON START AT 200
1538  001332  012706  000500          START:  MOV     #500,SP         ;SET STACK POINTER
1539  001336  013746  000004                  MOV     @#4,-(SP)       ;SAVE ERROR TRAP VECTOR
1540  001342  013746  000006                  MOV     @#6,-(SP)       ;AND VECTOR +2
1541  001346  012737  001372  000004         MOV     #1$,@#4         ;SET NEW VECTOR
1542  001354  005037  000006                  CLR     @#6             ;AND PSW
1543  001360  022777  177777  177202         CMP     #-1,@SWR        ;USE SOFTWARE SWITCH IF HARDWARE
1544  001366  001402                          BEQ     2$              ;IS = 177777
1545  001370  000404                          BR      3$              ;OTHERWISE USE HARDWARE SWR
1546  001372  022626                  1$:     CMP     (SP)+,(SP)+     ;RESET STACK PTR
1547  001374  012737  000176  000570  2$:     MOV     #SWREG,SWR      ;SET SOFTWARE SWITCH REGISTER
1548  001402  012637  000006          3$:     MOV     (SP)+,@#6       ;RESTORE ERROR TRAP VECTORS
1549  001406  012637  000004                  MOV     (SP)+,@#4
1550  001412  005037  001012                  CLR     SKAT            ;CLEAR SKIP ADDRESS TEST FLAG
1551  001416  005027                          CLR     (PC)+           ;;CLEAR CHAIN INDICATOR
1552  001420  000000          CHNFLG: .WORD   0               ;;CHAIN MODE INDICATOR
1553                                                          ;;1/0 = CHAIN/NOT CHAIN MODE
1554  001422  005737  000042                  TST     @#42            ;;BRANCH IF IN DUMP MODE
1555  001426  001407                          BEQ     50$
1556  001430  012737  000176  000570          MOV     #SWREG,SWR      ;;INVOKE SOFTWARE SWR
1557  001436  005237  001420                  INC     CHNFLG          ;;SET CHNFLG = CHAIN MODE
1558  001442  000137  001446                  JMP     SCHN            ;;GO TO CHAIN ADDRESS
1559  001446                          50$:
1560  001446  000240          SCHN:   NOP
1561  001450  122737  000006  000041  4$:     CMPB    #6,@#41         ;BRANCH IF NOT LOADED VIA TMDP
1562  001456  001005                          BNE     5$
1563  001460  012704  023364                  MOV     #MSG62,R4       ;ADVISE USER TO REMOVE TMDP FROM
1564  001464  004737  017662                  JSR     PC,TTOUT        ;UNIT UNDER TEST
1565  001470  000000                          HALT
1566  001472  012704  020600          5$:     MOV     #MSG1,R4
1567  001476  004737  017662                  JSR     PC,TTOUT        ;PRINT TITLE
1568  001502  005737  001420                  TST     CHNFLG          ;SEE IF IN CHAIN MODE
1569  001506  001402                          BEQ     6$              ;IF NOT: BR
1570  001510  000137  002222                  JMP     TSCD            ;ELSE GO TO START OF TESTS
1571  001514  112737  000043  020600  6$:     MOVB    #'#,MSG1        ;DO NOT PRINT TITLE ON RESTART
1572  001522  012704  022541                  MOV     #MSG44,R4
1573  001526  004737  017662                  JSR     PC,TTOUT        ;REQUEST REGISTER ADDRESS
1574  001532  013703  000606                  MOV     REGS,R3
1575  001536  004737  020010                  JSR     PC,OCTP         ;PRINT CURRENT ADDRESS
1576  001542  012705  000606                  MOV     #REGS,R5        ;SET ADDRESS SAVE LOC
1577  001546  012701  000007                  MOV     #7,R1           ;SET SIZE OF RESPONSE
1578  001552  012702  176400                  MOV     #176400,R2      ;SET UPPER LIMIT
1579  001556  012703  172300                  MOV     #172300,R3      ;SET LOWER LIMIT
1580  001562  004737  017340                  JSR     PC,TTR          ;GO GET RESPONSE
1581  001566  012704  022563                  MOV     #MSG45,R4
1582  001572  004737  017662                  JSR     PC,TTOUT        ;REQUEST VECTOR
1583  001576  013703  000604                  MOV     VECT,R3
1584  001602  004737  020010                  JSR     PC,OCTP         ;PRINT CURRENT VECTOR
1585  001606  012705  000604                  MOV     #VECT,R5        ;SET ADDRESS SAVE LOC
1586  001612  012701  000004                  MOV     #4,R1           ;SET SIZE OF RESPONSE
1587  001616  012702  000224                  MOV     #224,R2         ;SET UPPER LIMIT
1588  001622  012703  000150                  MOV     #150,R3         ;SET LOWER LIMIT
1589  001626  004737  017340                  JSR     PC,TTR          ;GO GET RESPONSE
```

```
 1590  001632  013700  000604                  MOV     VECT,R0         ;GET VECTOR
 1591  001636  012720  017202                  MOV     #MTINT,(R0)+    ;LOAD INTERRUPT ADDRESS IN VECTOR
 1592  001642  012710  000340                  MOV     #340,(R0)       ;LOAD PRIORITY
 1593  001646  013700  000606                  MOV     REGS,R0         ;GET START OF REGS
 1594  001652  012701  000016                  MOV     #16,R1          ;SET NUMBER OF REGS
 1595  001656  012702  000510                  MOV     #C1,R2          ;GET START OF TABLE
 1596  001662  010022                  ST0:    MOV     R0,(R2)+        ;BUILD TABLE
 1597  001664  062700  000002                  ADD     #2,R0           ;BUMP ADDRESS
 1598  001670  005301                          DEC     R1              ;SEE IF DONE
 1599  001672  001373                          BNE     ST0             ;IF NOT: BR
 1600  001674  012702  000610                  MOV     #T0B,R2
 1601  001700  012700  000077                  MOV     #77,R0
 1602  001704  005022                  ST1:    CLR     (R2)+           ;CLEAR FLAGS + COUNTERS
 1603  001706  005300                          DEC     R0
 1604  001710  001375                          BNE     ST1
 1605  001712  012704  023174                  MOV     #MS57A,R4       ;+REQUEST IF JUMPER IS IN NON-STANDARD MODE
 1606  001716  004737  017662                  JSR     PC,TTOUT        ;+
 1607  001722  012705  001016                  MOV     #JUMPER,R5      ;+
 1608  001726  012703  000000                  MOV     #0,R3           ;+LIMIT RESPONSE
 1609  001732  012701  000002                  MOV     #2,R1           ;+SET CHAR. NUMBER TO 1
 1610  001736  012702  000004                  MOV     #4,R2           ;+SET RANGE 0 - 4
 1611  001742  004737  017340                  JSR     PC,TTR          ;+GET RESPONSE
 1612  001746  022737  000002  001016          CMP     #2,JUMPER       ;TEST FOR NON-STANDARD JUMPER.
 1613  001754  001002                          BNE     1$
 1614  001756  004737  016624                  JSR     PC,NOST         ;GO TO MODIFY SCHEDLAR
 1615  001762  012704  023156          1$:     MOV     #MSG57,R4       ;REQUEST TM03 DRIVE #
 1616  001766  004737  017662                  JSR     PC,TTOUT
 1617  001772  013703  000620                  MOV     DRVN,R3         ;GET CURRENT DRIVE #
 1618  001776  004737  020010                  JSR     PC,OCTP         ;PRINT IT
 1619  002002  012705  000620                  MOV     #DRVN,R5        ;TTR ROUTINE RETURNS USER VALUE TO (R5)
 1620  002006  012701  000002                  MOV     #2,R1           ;LIMIT RESPONSE
 1621  002012  012702  000007                  MOV     #7,R2           ;LIMIT RANGE TO 0-7
 1622  002016  012703  000000                  MOV     #0,R3
 1623  002022  004737  017340                  JSR     PC,TTR          ;GET USER RESPONSE
 1624  002026  012704  023335                  MOV     #MSG58,R4       ;REQUEST TE16 SLAVE #
 1625  002032  004737  017662                  JSR     PC,TTOUT
 1626  002036  013703  000660                  MOV     SLVN,R3         ;GET CURRENT SLAVE #
 1627  002042  004737  020010                  JSR     PC,OCTP         ;AND PRINT IT
 1628  002046  012705  000660                  MOV     #SLVN,R5        ;TTR ROUTINE RETURNS RESPONSE TO (R5)
 1629  002052  012701  000002                  MOV     #2,R1           ;LIMIT RESONSE TO 1 CHARACTER
 1630  002056  012702  000007                  MOV     #7,R2           ;BETWEEN 0 AND 7
 1631  002062  012703  000000                  MOV     #0,R3
 1632  002066  004737  017340                  JSR     PC,TTR          ;GET USER RESPONSE
 1633  002072  012704  023131                  MOV     #MSG56,R4
 1634  002076  004737  017662                  JSR     PC,TTOUT        ;REQUEST STATIC ONLY
 1635  002102  013703  001006                  MOV     STATC,R3        ;GET CURRENT VALUE
 1636  002106  004737  020010                  JSR     PC,OCTP         ;AND TYPE IT
 1637  002112  012705  001006                  MOV     #STATC,R5       ;SET ADDRESS OF STATIC FLAG
 1638  002116  012701  000002                  MOV     #2,R1           ;SET SIZE OF RESPONSE
 1639  002122  012702  000001                  MOV     #1,R2           ;SET UPPER LIMIT
 1640  002126  012703  000000                  MOV     #0,R3           ;SET LOWER LIMIT
 1641  002132  004737  017340                  JSR     PC,TTR          ;GET RESPONSE
 1642
 1643  002136  012704  023510                  MOV     #MSG67,R4       ;++B REQUEST SLAVE TYPE 'TE16 OR TU77'
 1644  002142  004737  017662                  JSR     PC,TTOUT        ;++B
 1645  002146  013703  001010                  MOV     SLVTYP,R3       ;++B GET CURRENT SLAVE TYPE
```

```
1646  002152  004737  020010                  JSR     PC,OCTP         ;++B TYPE CURRENT VALUE
1647  002156  012705  001010                  MOV     #SLVTYP,R5      ;++B GET ADDRESS OF SLVTYP FLAG
1648  002162  012701  000002                  MOV     #2,R1           ;++B SET SIZE OF RESPONSE
1649  002166  012702  000001                  MOV     #1,R2           ;++B SET UPPER LIMIT
1650  002172  012703  000000                  MOV     #0,R3           ;++B SET LOWER LIMIT
1651  002176  004737  017340                  JSR     PC,TTR          ;++B GET RESPONSE
1652
1653                                  ;START 210
1654  002202  012706  000500          ST2:    MOV     #500,SP         ;SET STACK PTR
1655  002206  005037  001002                  CLR     RDRVF           ;CLEAR REVERSE FLAG
1656  002212  005037  001014                  CLR     PCNTR           ;CLEAR PASS COUNTER
1657  002216  004737  020436                  JSR     PC,GTSWR        ;GET SWITCHES
```

```
1658
1659                                        ;TEST SCHEDULAR**********
1660
1661  002222  052777  000100  176342  TSCD:   BIS     #100,@TKS       ;SET KEYBOARD INTERRUPT ENABLE
1662  002230  005037  000774                  CLR     WPGFL           ;CLEAR WRAP PATRN FLAG
1663  002234  005037  000734                  CLR     STFLG           ;CLEAR SINGLE TEST FLAG
1664  002240  017700  176324                  MOV     @SWR,R0
1665  002244  042700  177700                  BIC     #177700,R0      ;BRANCH IF SINGLE
1666  002250  001122                          BNE     STSCD           ;TEST SELECTED
1667  002252  005737  001420                  TST     CHNFLG          ;;BRANCH IF NOT IN CHAIN MODE
1668  002256  001457                          BEQ     TSCDA
1669  002260  012737  177777  000620          MOV     #-1,DRVN        ;;INITIALIZE DRIVE #
1670  002266  012737  177777  000660  NXTDRV: MOV     #-1,SLVN        ;;INITIALIZE SLAVE #
1671  002274  012777  000040  176216  1$:     MOV     #40,@CS         ;;INIT CONTROLLER
1672  002302  005237  000620                  INC     DRVN            ;;STEP DRIVE #
1673  002306  022737  000010  000620          CMP     #10,DRVN        ;;EXIT IF ALL DRIVES TESTED
1674  002314  001521                          BEQ     $DONE           ;;FOR AVAILABILITY
1675  002316  013777  000620  176174          MOV     DRVN,@CS        ;;LOAD DRIVE #
1676  002324  005777  176160                  TST     @C1             ;;ACCESS DRIVE
1677  002330  032777  010000  176162          BIT     #10000,@CS      ;;BRANCH IF DRIVE NON EXISTANT
1678  002336  001356                          BNE     1$              ;;(NED = 1)
1679  002340  005237  000660          NXTSLV: INC     SLVN            ;;STEP SLAVE # AND BRANCH
1680  002344  001011                          BNE     1$              ;;IF NOT SLAVE 0
1681  002346  005737  000620                  TST     DRVN            ;;BRANCH IF NOT DRIVE # 0
1682  002352  001006                          BNE     1$
1683  002354  122737  000006  000041          CMPB    #6,@#41         ;;BRANCH IF NOT TMDP
1684  002362  001002                          BNE     1$
1685  002364  005237  000660                  INC     SLVN            ;;STEP TO SLAVE # 1
1686  002370  022737  000010  000660  1$:     CMP     #10,SLVN        ;;BRANCH IF ALL SLAVES TESTED
1687  002376  001733                          BEQ     NXTDRV          ;;FOR AVAILABILITY
1688  002400  013777  000660  176134          MOV     SLVN,@TC        ;;LOAD SLAVE UNIT #
1689  002406  032777  002000  176122          BIT     #2000,@DT       ;;BRANCH IF SLAVE NOT
1690  002414  001751                          BEQ     NXTSLV          ;;PRESENT (SPR = 0)
1691  002416  012737  001056  000736  TSCDA:  MOV     #TSTTBL,LTADD
1692  002424  062737  000004  000736  TSCD0:  ADD     #4,LTADD
1693  002432  013737  000736  000710  TSCD1:  MOV     LTADD,ITRLP
1694  002440  062737  000002  000710          ADD     #2,ITRLP        ;SET ITERATION ADDRESS
1695  002446  005037  000614                  CLR     HDRFL           ;CLEAR PRINT HEADER FLAG
1696  002452  017700  176260                  MOV     @LTADD,R0       ;SET POINTER TO TEST
1697  002456  000110                          JMP     (R0)            ;GO TO TEST
1698  002460  032777  002000  176102  TSCD2:  BIT     #2000,@SWR      ;SEE IF HALT ON TEST
1699  002466  001403                          BEQ     TSCD3           ;IF NOT: BR
1700  002470  000000                          HALT
1701  002472  005037  000774                  CLR     WPGFL           ;CLEAR WRAP DATA GENERATOR FLAG
1702  002476  005737  000734          TSCD3:  TST     STFLG           ;SE IF SINGLE TEST
1703  002502  001750                          BEQ     TSCD0           ;IF NOT: BR
1704  002504  017700  176060                  MOV     @SWR,R0
1705  002510  042700  177700                  BIC     #177700,R0      ;BRANCH IF ALL TESTS DESIRED
1706  002514  001642                          BEQ     TSCD            ;IF SO: BR
1707  002516  012737  000001  000734  STSCD:  MOV     #1,STFLG        ;SET SINGLE TEST FLAG
1708  002524  023700  001330                  CMP     TLAST,R0        ;SEE IF EXCEEDED TESTS
1709  002530  002410                          BLT     TEND            ;IF SO: BR
1710  002532  006300                          ASL     R0
1711  002534  006100                          ROL     R0              ;SET TABLE MODIFIER
1712  002536  012737  001056  000736          MOV     #TSTTBL,LTADD
1713  002544  060037  000736                  ADD     R0,LTADD        ;SET TEST POINTER
```

```
1714  002550  000730                          BR      TSCD1
1715  002552  005737  001420          TEND:   TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
1716  002556  001270                          BNE     NXTSLV          ;STEP TO NEXT SLAVE
1717  002560  012704  022401          $DONE:  MOV     #MSG41,R4
1718  002564  004737  017662                  JSR     PC,TTOUT        ;PRINT END OF PASS
1719  002570  013703  001014                  MOV     PCNTR,R3
1720  002574  004737  020010                  JSR     PC,OCTP         ;PRINT PASS NUMBER
1721  002600  005000                          CLR     R0              ;DELAY WAITING FOR
1722  002602  005300                  1$:     DEC     R0
1723  002604  001376                          BNE     1$
1724  002606  013700  000042                  MOV     @#42,R0         ;GET ACT11 RETURN ADDRESS
1725  002612  001405                          BEQ     HERE            ;BRANCH IF NOT ACT11
1726  002614  000005                          RESET
1727  002616  004710                  $ENDAD: JSR     PC,(R0)
1728  002620  000240                          NOP
1729  002622  000240                          NOP
1730  002624  000240                          NOP
1731  002626  000240                  HERE:   NOP
1732  002630  005737  001420                  TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
1733  002634  001005                          BNE     TENDX
1734  002636  032777  010000  175724          BIT     #10000,@SWR     ;SEE IF HALT ON PASS
1735  002644  001001                          BNE     TENDX           ;IF NOT: BR
1736  002646  000000                          HALT
1737  002650  012737  000001  001012  TENDX:  MOV     #1,SKAT         ;SET SKIP ADDRESS TEST FLAG
1738  002656  005237  001014                  INC     PCNTR           ;BUMP PASS COUNTER
1739  002662  000137  002222                  JMP     TSCD            ;RESTART
1740                                          ;LOGIC TEST 1: DRIVE ADDRESSING**********
1741
1742  002666  012737  023630  000616  LT1:    MOV     #MSLT1,EMADDR   ;++B SET ERROR MSG HDR ADDRESS
1743  002674  013737  000620  000674          MOV     DRVN,TEMP3      ;GET DRIVE # TO BE TESTED
1744  002702  013701  000620                  MOV     DRVN,R1
1745  002706  005737  001012                  TST     SKAT            ;SEE IF SKIP ADDRESS TESTS
1746  002712  001403                          BEQ     1$              ;IF NOT: BR
1747  002714  005737  000734                  TST     STFLG           ;SEE IF SINGLE TEST
1748  002720  001506                          BEQ     LT1X            ;IF NOT: BR
1749  002722  032777  001000  175640  1$:     BIT     #1000,@SWR      ;BRANCH IF MAN INTERVENTION
1750  002730  001430                          BEQ     LT1A            ;NOT SELECTED
1751  002732  012704  021044          LT1G0:  MOV     #MSG2A,R4
1752  002736  004737  017134                  JSR     PC,INST         ;PRINT TEST INSTRUCTIONS
1753  002742  012704  021003          LT1G:   MOV     #MSG2,R4
1754  002746  004737  017662                  JSR     PC,TTOUT        ;REQUEST DRIVE NUMBER
1755  002752  012705  000674                  MOV     #TEMP3,R5       ;TTR ROUTINE RETURNS RESPONSE TO (R5)
1756  002756  012701  000002                  MOV     #2,R1
1757  002762  012702  000007                  MOV     #7,R2
1758  002766  012703  000000                  MOV     #0,R3
1759  002772  004737  017340                  JSR     PC,TTR          ;GET DRIVE NUMBER
1760  002776  005737  000670                  TST     TEMP1           ;SEE IF ANOTHER DRIVE
1761  003002  001455                          BEQ     LT1X            ;IF NOT: BR
1762  003004  005001                          CLR     R1              ;SELECT DRIVE 0
1763  003006  012700  000010                  MOV     #10,R0          ;SET NUMBER OF DRIVES
1764  003012  012777  000040  175500  LT1A:   MOV     #40,@CS         ;INIT
1765  003020  010177  175474                  MOV     R1,@CS          ;SELECT DRIVE
1766  003024  005777  175460                  TST     @CS             ;ACCESS DRIVE
1767  003030  032777  010000  175462          BIT     #10000,@CS      ;SEE IF NED
1768  003036  001010                          BNE     LT1B            ;IF SO: BR
1769  003040  032777  001000  175522          BIT     #1000,@SWR      ;BRANCH IF NOT MANUAL INTERVENTION
```

```
1770  003046  001433                          BEQ     LT1X
1771  003050  023701  000674                  CMP     TEMP3,R1        ;SEE IF SHOULD BE NED
1772  003054  001404                          BEQ     LT1C            ;IF NOT: BR
1773  003056  000407                          BR      LT1ER           ;ELSE GO TO ERROR
1774  003060  023701  000674          LT1B:   CMP     TEMP3,R1        ;SEE IF SHOULD BE NED
1775  003064  001410                          BEQ     LT1ER1
1776  003066  005300                  LT1C:   DEC     R0
1777  003070  001724                          BEQ     LT1G            ;IF DONE ALL: BR
1778  003072  005201                          INC     R1              ;SELECT NEXT DRIVE
1779  003074  000746                          BR      LT1A            ;CONTINUE
1780  003076  012737  000001  000712  LT1ER:  MOV     #1,EXFL         ;FLAG EXPT
1781  003104  000403                          BR      LT1ER2
1782  003106  012737  000002  000712  LT1ER1: MOV     #2,EXFL         ;FLAG NOT EXPT
1783  003114  012737  021173  000666  LT1ER2: MOV     #MSG3,ERADD     ;FLAG CONDITION
1784  003122  012737  003012  000706          MOV     #LT1A,SCOLP     ;SET SCOPE ADDRESS
1785  003130  004737  015164                  JSR     PC,LTGER        ;GO PRINT LOGIC TEST ERROR
1786  003134  000754                          BR      LT1C            ;CONTINUE TEST
1787  003136  000137  002460          LT1X:   JMP     TSCD2           ;RETURN TO SCHED
1788
```

```
1789                                            ;LCGIC TEST 2: REGISTER ADDRESSING**********
1790
1791  003142  000240                  LT2:      NOP
1792  003144  012777  000040  175346  LT2IT:    MOV     #40,@CS         ;INIT
1793  003152  013777  000620  175340            MOV     DRVN,@CS        ;SELECT DRIVE
1794  003160  012737  023704  000616            MOV     #MSLT2,EMADDR   ;SAVE LT2 HEADER ADDRESS
1795  003166  012705  000510                    MOV     #C1,R5          ;SET ADDRESS OF FIRST REGISTER
1796  003172  012700  000016                    MOV     #16,R0          ;SET NUMBER OF REGISTERS
1797  003176  012702  000622                    MOV     #TR00,R2        ;SET START OF REGISTER BUFFER
1798  003202  011501                  LT2A:     MOV     (R5),R1
1799  003204  011112                            MOV     (R1),(R2)       ;READ REGISTER
1800  003206  032777  020000  175274            BIT     #20000,@C1      ;SEE IF ERROR
1801  003214  001402                            BEQ     LT2B            ;IF NOT: BR
1802  003216  004737  003246                    JSR     PC,LT2ER1       ;ELSE GO TO ERROR 1
1803  003222  032777  000002  175274  LT2B:     BIT     #2,@ER          ;SEE IF ILR
1804  003230  001402                            BEQ     LT2C            ;IF NOT: BR
1805  003232  004737  003264                    JSR     PC,LT2ER2       ;ELSE GO TO ERROR 2
1806  003236  022225                  LT2C:     CMP     (R2)+,(R5)+     ;BUMP ADDRESS
1807  003240  005300                            DEC     R0
1808  003242  001357                            BNE     LT2A            ;CONTINUE FOR ALL REGISTERS
1809  003244  000434                            BR      LT2X
1810
1811  003246  012737  000002  000712  LT2ER1:  MOV     #2,EXFL         ;FLAG NOT EXPECTED
1812  003254  012737  021215  000666            MOV     #MSG4,ERADD     ;POINT TO CONTROLLER ERROR
1813  003262  000415                            BR      LT2ERG          ;GO TO ERROR
1814  003264  012737  000002  000712  LT2ER2:  MOV     #2,EXFL         ;FLAG NOT EXPECTED
1815  003272  012737  021233  000666            MOV     #MSG5,ERADD     ;POINT TO DRIVE ERROR
1816  003300  000406                            BR      LT2ERG          ;GO TO ERROR
1817  003302  012737  000001  000712  LT2ER3:  MOV     #1,EXFL         ;FLAG EXPECTED
1818  003310  012737  021215  000666            MOV     #MSG4,ERADD     ;POINT TO DRIVE
1819  003316  012737  003332  000706  LT2ERG:  MOV     #LT2LP,SCOLP    ;SET SCOPE ADDRESS
1820  003324  004737  015164                    JSR     PC,LTGER        ;GO PRINT
1821  003330  000207                            RTS     PC              ;ELSE CONTINUE
1822  003332  005726                  LT2LP:    TST     (SP)+           ;RESET STACK
1823  003334  000722                            BR      LT2A            ;LOOP
1824  003336  004737  016534          LT2X:     JSR     PC,ITER         ;GO SEE IF ITERATIONS
1825  003342  000137  002460                    JMP     TSCD2           ;RETURN TO SCHED
```

```
1826                                        ;LOGIC TEST 3: CONTROL BUS**********
1827
1828  003346  000240                LT3:    NOP
1829  003350  012737  023763  000616 LT3IT:  MOV     #MSLT3,EMADDR   ;SET TEST HEADER
1830  003356  012701  000001                MOV     #1,R1           ;PRESET PATTERN 1
1831  003362  012700  000020                MOV     #20,R0          ;SET PATTERN CHANGE NUMBER
1832  003366  004737  016662        LT3A:   JSR     PC,INIT1        ;GO INIT
1833  003372  010177  175120                MOV     R1,@FC          ;WRITE TO FC
1834  003376  032777  000010  175120        BIT     #10,@ER         ;SEE IF CPAR (TM03)
1835  003404  001012                        BNE     LT3ER1          ;IF SO: BR
1836  003406  017702  175104        LT3B:   MOV     @FC,R2          ;READ FC
1837  003412  032777  020000  175070        BIT     #20000,@C1      ;SEE IF MCPE (RH)
1838  003420  001017                        BNE     LT3ER2          ;IF SO: BR
1839  003422  005300                LT3C:   DEC     R0              ;SEE IF DONE PATTERN CHANGES
1840  003424  001426                        BEQ     LT3X            ;IF SO: BR
1841  003426  006301                        ASL     R1              ;CHANGE PATTERN
1842  003430  000756                        BR      LT3A            ;CONTINUE
1843  003432  012737  021546  000666 LT3ER1: MOV     #MSG11,ERADD    ;SET ERROR CODE
1844  003440  012737  003366  000706        MOV     #LT3A,SCOLP     ;SET SCOPE ADDRESS
1845  003446  017702  175044                MOV     @FC,R2          ;GET DATA
1846  003452  004737  016272                JSR     PC,LTGER1       ;GO DO ERROR
1847  003456  000753                        BR      LT3B
1848  003460  012737  021522  000666 LT3ER2: MOV     #MSG10,ERADD    ;SET ERROR CODE
1849  003466  012737  003406  000706        MOV     #LT3B,SCOLP     ;SET SCOPE ADDRESS
1850  003474  004737  016272                JSR     PC,LTGER1       ;GO DO ERROR
1851  003500  000750                        BR      LT3C
1852  003502  105701                LT3X:   TSTB    R1              ;SEE IF DONE PATTERN 2
1853  003504  100405                        BMI     LT3XX           ;IF SO: BR
1854  003506  012701  000401                MOV     #401,R1         ;SET PATTERN 2
1855  003512  012700  000010                MOV     #10,R0          ;SET PATTERN CHANGE NUMBER
1856  003516  000723                        BR      LT3A            ;DO PATTERN 2
1857  003520  004737  016534        LT3XX:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
1858  003524  000137  002460                JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
1859
1860                                        ;LOGIC TEST 4: SLAVE ADDRESSING**********
1861
1862  003530  013737  000660  000674  LT4:  MOV   SLVN,TEMP3
1863  003536  013701  000660              MOV   SLVN,R1
1864  003542  005737  001012              TST   SKAT            ;SEE IF SKIP ADDRESS TESTS
1865  003546  001403                      BEQ   1$              ;IF NOT: BR
1866  003550  005737  000734              TST   STFLG           ;SEE IF SINGLE TEST
1867  003554  001564                      BEQ   LT4X            ;IF NOT: BR
1868  003556  032777  001000  175004  1$: BIT   #1000,@SWR      ;BRANCH IF MAN INTERVETION
1869  003564  001430                      BEQ   LT4A            ;NOT SELECTED
1870  003566  012704  021351        LT4G0: MOV  #MSG8A,R4
1871  003572  004737  017134              JSR   PC,INST         ;PRINT TEST INSTRUCTIONS
1872  003576  012704  021310        LT4G: MOV   #MSG8,R4
1873  003602  004737  017662              JSR   PC,TTOUT        ;REQUEST SLAVE
1874  003606  012705  000674              MOV   #TEMP3,R5
1875  003612  012701  000002              MOV   #2,R1
1876  003616  012702  000007              MOV   #7,R2
1877  003622  012703  000000              MOV   #0,R3
1878  003626  004737  017340              JSR   PC,TTR          ;GET SLAVE NUMBER
1879  003632  005737  000670              TST   TEMP1           ;SEE IF SLAVE
1880  003636  001533                      BEQ   LT4X            ;IF NOT: BR
1881  003640  005001                      CLR   R1              ;SELECT SLAVE 0
1882  003642  012700  000010              MOV   #10,R0          ;SET NUMBER OF SLAVES
1883  003646  012777  000040  174644  LT4A: MOV  #40,@CS        ;INIT
1884  003654  013777  000620  174636      MOV   DRVN,@CS        ;SELECT DRIVE
1885  003662  010177  174654              MOV   R1,@TC          ;SELECT SLAVE
1886  003666  017703  174644              MOV   @DT,R3          ;GET DT
1887  003672  020137  000674              CMP   R1,TEMP3        ;SEE IF SHOULD HAVE SPR
1888  003676  001404                      BEQ   LT4B            ;IF SO: BR
1889  003700  032703  002000              BIT   #2000,R3        ;SEE IF SPR
1890  003704  001461                      BEQ   LT4D            ;IF NOT: BR
1891  003706  000464                      BR    LT4ER1          ;GO TO ERROR 1
1892  003710  032703  002000        LT4B: BIT   #2000,R3        ;SEE IF NO SLAVE PRESENT
1893  003714  001465                      BEQ   LT4ER2          ;(SPR=0)
1894  003716  012704  023460        LT4C: MOV   #MSG64,R4       ;TYPE 'SLAVE TYPE = '
1895  003722  004737  017662              JSR   PC,TTOUT
1896  003726  012702  000001              MOV   #1,R2           ;PRESET SLAVE TYPE - TU77
1897  003732  012704  023476              MOV   #MSG65,R4       ;SET UP TO TYPE 'TU77'
1898  003736  022777  142054  174572      CMP   #142054,@DT     ;BRANCH IF TU77
1899  003744  001412                      BEQ   1$
1900  003746  012704  023503              MOV   #MSG66,R4       ;CHANGE SLAVE TYPE TO 'TE16'
1901  003752  005302                      DEC   R2              ;CHANGE SLAVE TYPE TO TE16
1902  003754  022777  142051  174554      CMP   #142051,@DT     ;BRANCH IF TE16
1903  003762  001403                      BEQ   1$
1904  003764  005302                      DEC   R2              ;CHENGE SLAVE TYPE TO ILLEGAL
1905  003766  012704  023620              MOV   #MSG69,R4
1906  003772  004737  017662        1$:   JSR   PC,TTOUT        ;TYPE SLAVE TYPE (TU77,TE16, OR ILLEGAL)
1907  003776  020237  001010              CMP   R2,SLVTYP       ;BRANCH IF HARDWARE SLAVE TYPE IS THE
1908  004002  001406                      BEQ   4$              ;SAME AS USER SPACIFIED SLAVE TYPE
1909  004004  012704  023546              MOV   #MSG68,R4       ;++B TYPE 'INCORRECT SLAVE TYPE'
1910  004010  004737  017662              JSR   PC,TTOUT        ;++B
1911  004014  000137  002552              JMP   TEND            ;++B EXIT TEST
1912  004020  012704  022223        4$:   MOV   #MSG30,R4
1913  004024  004737  017662              JSR   PC,TTOUT        ;PRINT SERIAL NUMBER TAG
1914  004030  017703  174504              MOV   @SN,R3
```

```
1915  004034  004737  020334                 JSR     PC,SNPT          ;PRINT SERIAL NUMBER
1916  004040  032777  001000  174522         BIT     #1000,@SWR       ;BRANCH IF NOT MANUAL INTERVENTION
1917  004046  001427                         BEQ     LT4X
1918  004050  005300                 LT4D:   DEC     R0
1919  004052  001651                         BEQ     LT4G             ;IF DONE ALL: BR
1920  004054  005201                         INC     R1               ;BUMP SLAVE
1921  004056  000673                         BR      LT4A             ;CONTINUE
1922  004060  012737  000001  000712 LT4ER1: MOV     #1,EXFL          ;FLAG EXPT: NOT RECEIVED
1923  004066  000403                         BR      LT4ERG
1924  004070  012737  000002  000712 LT4ER2: MOV     #2,EXFL          ;FLAG RECVD: NOT EXPT
1925  004076  012737  024050  000616 LT4ERG: MOV     #MSLT4,EMADDR    ;SET  LT4 HEADER
1926  004104  012737  021500  000666         MOV     #MSG9,ERADD      ;SET ERROR CONDITION
1927  004112  012737  003646  000706         MOV     #LT4A,SCOLP      ;SET SCOPE ADDRESS
1928  004120  004737  015164                 JSR     PC,LTGER         ;GO TO ERROR
1929  004124  000751                         BR      LT4D             ;IF NO SCOPE: BR
1930  004126  000137  002460         LT4X:   JMP     TSCD2            ;RETURN TO SCHED
1931
```

```
1932                                            ;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST**********
1933
1934  004132  012737  024132  000616  LT5:    MOV     #MSLT5,EMADDR   ;SET TEST HEADER
1935  004140  004737  016662          LT5IT:  JSR     PC,INIT1        ;GO INIT
1936  004144  012700  000032                  MOV     #32,R0          ;SET LOOP FOR BITS 4-0
1937  004150  005001                          CLR     R1              ;SET TEST WORD
1938  004152  010177  174356          LT5A:   MOV     R1,@MR          ;SEND TEST WORD TO MR
1939  004156  017702  174352                  MOV     @MR,R2          ;READ MR
1940  004162  042702  177740                  BIC     #177740,R2      ;MASK BITS 4-0
1941  004166  020102                          CMP     R1,R2           ;SEE IF EXPT = RECVD
1942  004170  001026                          BNE     LT5ER1
1943  004172  005300          LT5B:   DEC     R0
1944  004174  001402                          BEQ     LT5C            ;IF DONE LOOP: BR
1945  004176  005201                          INC     R1              ;BUMP TEST WORD
1946  004200  000764                          BR      LT5A            ;CONTINUE LOOP
1947  004202  012701  000015          LT5C:   MOV     #15,R1          ;SET TEST WORD + WAM 3
1948  004206  012700  001000                  MOV     #1000,R0        ;SET LOOP FOR BITS 15-7
1949  004212  010177  174316          LT5D:   MOV     R1,@MR          ;LOAD MR
1950  004216  017702  174312                  MOV     @MR,R2          ;READ MR
1951  004222  042702  000140                  BIC     #140,R2         ;MASK OUT BITS 5,6
1952  004226  020102                          CMP     R1,R2           ;SEE IF EXPT = RECVD
1953  004230  001401                          BEQ     LT5E            ;IF SO: BR
1954  004232  000416                          BR      LT5ER2          ;ELSE GO TO ERR 2
1955  004234  005300          LT5E:   DEC     R0
1956  004236  001425                          BEQ     LT5X            ;IF DONE LOOP: BR
1957  004240  062701  000200                  ADD     #200,R1         ;BUMP TEST WORD
1958  004244  000762                          BR      LT5D            ;CONTINUE LOOP
1959  004246  012737  021611  000666  LT5ER1: MOV     #MSG14,ERADD    ;SET ERROR CODE
1960  004254  012737  004152  000706          MOV     #LT5A,SCOLP     ;SET SCOPE ADDRESS
1961  004262  004737  016272                  JSR     PC,LTGER1       ;GO TO ERROR
1962  004266  000741                          BR      LT5B            ;CONTINUE
1963  004270  012737  021626  000666  LT5ER2: MOV     #MSG15,ERADD    ;SET ERROR CODE
1964  004276  012737  004212  000706          MOV     #LT5D,SCOLP     ;SET SCOPE ADDRESS
1965  004304  004737  016272                  JSR     PC,LTGER1       ;GO TO ERROR
1966  004310  000751                          BR      LT5E            ;CONTINUE
1967  004312  004737  016534          LT5X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
1968  004316  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
1969
```

```
1970                                                ;LOGIC TEST 6: TC REGISTER BIT TEST**********
1971
1972  004322  000240                        LT6:    NOP
1973  004324  012737  024201  000616        LT6IT:  MOV     #MSLT6,EMADDR   ;POINT TO LT6 HEADER
1974  004332  012700  000003                        MOV     #3,R0           ;SET NUMBER OF TESTS
1975  004336  005001                        LT6A1:  CLR     R1
1976  004340  004737  016662                LT6A:   JSR     PC,INIT1        ;GO INIT
1977  004344  010177  174172                LT6B:   MOV     R1,@TC          ;WRITE TC
1978  004350  017702  174166                        MOV     @TC,R2          ;READ TC
1979  004354  042702  160000                        BIC     #160000,R2      ;MASK OUT SAC
1980  004360  020102                                CMP     R1,R2           ;SEE IF EXPT = RECVD
1981  004362  001010                                BNE     LT6ER1          ;IF NOT: BR
1982  004364  005300                        LT6D:   DEC     R0
1983  004366  001417                                BEQ     LT6X            ;IF DONE ALL: BR
1984  004370  022700  000001                        CMP     #1,R0           ;SEE IF RESET TEST
1985  004374  001760                                BEQ     LT6A1           ;IF SO: BR
1986  004376  012701  017777                        MOV     #17777,R1       ;SET TEST WORD
1987  004402  000756                                BR      LT6A            ;DO SET TEST
1988  004404  012737  021654  000666        LT6ER1: MOV     #MSG18,ERADD    ;SET ERROR CODE
1989  004412  012737  004344  000706                MOV     #LT6B,SCOLP     ;SET SCOPE ADDRES
1990  004420  004737  016272                        JSR     PC,LTGER1       ;GO TO ERROR
1991  004424  000757                                BR      LT6D            ;CONTINUE
1992  004426  004737  016534                LT6X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
1993  004432  000137  002460                        JMP     TSCD2           ;RETURN TO SCHED
1994
```

```
1995                                        ;LOGIC TEST 7: FRAME COUNT BIT TEST**********
1996
1997  004436  000240                LT7:    NOP
1998  004440  012700  000003        LT7IT:  MOV     #3,R0               ;SET TEST NUMBER
1999  004444  012737  024250  000616 LT7C:  MOV     #MSLT7,EMADDR       ;SET TEST HEADER
2000  004452  005001                        CLR     R1                  ;SET TEST WORD
2001  004454  004737  016662        LT7A:   JSR     PC,INIT1            ;GO INIT
2002  004460  010177  174032                MOV     R1,@FC              ;CLEAR FRAME COUNT
2003  004464  017702  174026                MOV     @FC,R2              ;READ FC
2004  004470  020102                        CMP     R1,R2               ;SEE IF EXPT = RECVD
2005  004472  001010                        BNE     LT7ER1
2006  004474  005300                LT7B:   DEC     R0                  ;SEE IF DONE ALL
2007  004476  001417                        BEQ     LT7X                ;IF SO: BR
2008  004500  022700  000001                CMP     #1,R0               ;SEE IF RESET TEST
2009  004504  001757                        BEQ     LT7C                ;IF SO: BR
2010  004506  012701  177777                MOV     #-1,R1              ;SET TEST WORD TO -1
2011  004512  000760                        BR      LT7A                ;CONTINUE
2012  004514  012737  021673  000666 LT7ER1: MOV     #MSG19,ERADD        ;SET ERROR CODE
2013  004522  012737  004454  000706        MOV     #LT7A,SCOLP         ;SET SCOPE ADDRESS
2014  004530  004737  016272                JSR     PC,LTGER1           ;GO PRINT ERROR
2015  004534  000757                        BR      LT7B                ;ELSE CONTINUE
2016  004536  012700  000003        LT7X:   MOV     #3,R0               ;RESET TEST AMT
2017  004542  004737  016534                JSR     PC,ITER             ;GO SEE IF ITERATIONS
2018  004546  000137  002460                JMP     TSCD2               ;RETURN TO SCHED
2019
```

```
2020                                        ;LOGIC TEST 10: FUNCTION CODE BIT TEST**********
2021
2022  004552  000240                LT10:   NOP
2023  004554  012737  024317  000616 LT10IT: MOV     #MSLT10,EMADDR  ;SET TEST HEADER
2024  004562  012700  000003              MOV     #3,R0           ;SET NUMBER OF TESTS
2025  004566  005001                LT10A1: CLR     R1              ;SET TEST WORRD
2026  004570  012777  000040  173722 LT10A:  MOV     #40,@CS         ;INIT
2027  004576  013777  000620  173714        MOV     DRVN,@CS        ;SELECT DRIVE
2028  004604  010177  173700              MOV     R1,@C1          ;WRITE C1
2029  004610  017702  173674              MOV     @C1,R2          ;READ C1
2030  004614  042702  177701              BIC     #177701,R2      ;MASK FUNCTION CODE
2031  004620  020102                      CMP     R1,R2           ;SEE IF EXPT = RECVD
2032  004622  001010                      BNE     LT10E1
2033  004624  005300                LT10B:  DEC     R0
2034  004626  001417                      BEQ     LT10X           ;IF DONE ALL: BR
2035  004630  022700  000001              CMP     #1,R0           ;SEE IF RESET TEST
2036  004634  001754                      BEQ     LT10A1          ;IF SO: BR
2037  004636  012701  000076              MOV     #76,R1          ;SET TEST WORD
2038  004642  000752                      BR      LT10A           ;DO SET TEST
2039  004644  012737  021712  000666 LT10E1: MOV     #MSG20,ERADD    ;SET ERROR CODE
2040  004652  012737  004570  000706        MOV     #LT10A,SCOLP    ;SET SCOPE ADDRESS
2041  004660  004737  016272              JSR     PC,LTGER1       ;GO PRINT ERROR
2042  004664  000757                      BR      LT10B           ;ELSE CONTINUE
2043  004666  004737  016534        LT10X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2044  004672  000137  002460              JMP     TSCD2           ;RETURN TO SCHED
```

```
2045
2046                                        ;LOGIC TEST 11: GO BIT SET RESET**********
2047
2048  004676  000240                LT11:   NOP
2049  004700  012737  024375  000616 LT11IT: MOV    #MSLT11,EMADDR  ;SET TEST HEADER
2050  004706  004737  016662               JSR    PC,INIT1        ;GO INIT
2051  004712  017702  173572               MOV    @C1,R2          ;READ C1
2052  004716  032702  000001               BIT    #1,R2           ;SEE IF GO=0
2053  004722  001030                       BNE    LT11E1
2054  004724  012777  000015  173602 LT11B: MOV    #15,@MR         ;SELECT WAM 3
2055  004732  005077  173560               CLR    @FC             ;ASSURE FCS = 1
2056  004736  052777  001700  173576       BIS    #1700,@TC       ;ASSURE FMT OK
2057  004744  012777  000071  173536       MOV    #71,@C1         ;SET READ+GO
2058  004752  017702  173532               MOV    @C1,R2          ;READ C1
2059  004756  032702  000001               BIT    #1,R2           ;SEE IF GO =1
2060  004762  001424                       BEQ    LT11E2
2061  004764  004737  016662        LT11C: JSR    PC,INIT1        ;GO INIT
2062  004770  017702  173514               MOV    @C1,R2          ;READ C1
2063  004774  032702  000001               BIT    #1,R2           ;SEE IF GO=0
2064  005000  001444                       BEQ    LT11X           ;IF SO:BR
2065  005002  000430                       BR     LT11E3          ;ELSE GO TO ERROR 3
2066  005004  012737  021744  000666 LT11E1: MOV    #MSG21,ERADD    ;SET ERROR CODE
2067  005012  012702  000001               MOV    #1,R2           ;SET REVD
2068  005016  005001                       CLR    R1              ;SET EXPT
2069  005020  012737  004700  000706       MOV    #LT11IT,SCOLP           ;SET SCOPE ADDRESS
2070  005026  004737  016272               JSR    PC,LTGER1       ;GO PRINT ERROR
2071  005032  000734                       BR     LT11B           ;ELSE CONTINUE
2072  005034  012737  022002  000666 LT11E2: MOV    #MSG22,ERADD    ;SET ERROR CODE
2073  005042  005002                       CLR    R2              ;SET RCVD
2074  005044  012701  000001               MOV    #1,R1           ;SET EXPT
2075  005050  012737  004724  000706       MOV    #LT11B,SCOLP    ;SET SCOPE ADDRESS
2076  005056  004737  016272               JSR    PC,LTGER1       ;GO PRINT ERROR
2077  005062  000740                       BR     LT11C           ;ELSE CONTINUE
2078  005064  012737  022023  000666 LT11E3: MOV    #MSG23,ERADD    ;SET ERROR CODE
2079  005072  005001                       CLR    R1              ;SET EXPT
2080  005074  012702  000001               MOV    #1,R2           ;SET RCVD
2081  005100  012737  004764  000706       MOV    #LT11C,SCOLP    ;SET SCOPE ADDRESS
2082  005106  004737  016272               JSR    PC,LTGER1       ;GO PRINT ERROR
2083  005112  004737  016534        LT11X: JSR    PC,ITER         ;GO SEE IF ITERATIONS
2084  005116  000137  002460               JMP    TSCD2           ;RETURN TO SCHED
```

```
2085
2086                                        ;LOGIC TEST 12: DRIVE READY BIT**********
2087
2088  005122  000240                LT12:   NOP
2089  005124  012737  024442  000616 LT12IT: MOV     #MSLT12,EMADDR  ;SET TEST HEADER
2090  005132  004737  016662                JSR     PC,INIT1        ;GO INIT
2091  005136  032777  000200  173356        BIT     #200,@DS        ;SEE IF DRY=1
2092  005144  001426                       BEQ     LT12E1
2093  005146  012777  000015  173360 LT12B:  MOV     #15,@MR         ;SET WAM3
2094  005154  005077  173336                CLR     @FC             ;ASSURE FCS = 1
2095  005160  052777  001700  173354        BIS     #1700,@TC       ;ASSURE FMT OK
2096  005166  012777  000071  173314        MOV     #71,@C1         ;SET READ+GO
2097  005174  032777  000200  173320        BIT     #200,@DS        ;SEE IF DRY=0
2098  005202  001020                       BNE     LT12E2
2099  005204  004737  016662        LT12C:  JSR     PC,INIT1        ;GO INIT
2100  005210  032777  000200  173304        BIT     #200,@DS        ;SEE IF DRY=1
2101  005216  001033                       BNE     LT12X           ;IF SO: BR
2102  005220  000422                       BR      LT12E3          ;ELSE GO TO ERROR 3
2103  005222  012737  022056  000666 LT12E1: MOV     #MSG24,ERADD    ;SET ERROR CODE
2104  005230  012737  005124  000706        MOV     #LT12IT,SCOLP   ;SET SCOPE ADDRESS
2105  005236  004737  016264                JSR     PC,LTGER2       ;GO TO ERROR
2106  005242  000741                       BR      LT12B           ;CONTINUE
2107  005244  012737  022104  000666 LT12E2: MOV     #MSG25,ERADD    ;SET ERROR CODE
2108  005252  012737  005146  000706        MOV     #LT12B,SCOLP    ;SET LOOP ADDRESS
2109  005260  004737  016264                JSR     PC,LTGER2       ;GO PRINT ERROR
2110  005264  000747                       BR      LT12C           ;CONTINUE
2111  005266  012737  022133  000666 LT12E3: MOV     #MSG25A,ERADD   ;SET ERROR CODE
2112  005274  012737  005204  000706        MOV     #LT12C,SCOLP    ;SET ERROR LOOP
2113  005302  004737  016264                JSR     PC,LTGER2       ;GET PRINT ERROR
2114  005306  004737  016534        LT12X:  JSR     PC,ITER         ;GO TO ITERATION SUBROUTINE
2115  005312  000137  002460                JMP     TSCD2           ;RETURN TO SCHED
```

```
2116
2117                                        ;LOGIC TEST 13: INTERRUPT TEST**********
2118
2119  005316  005000                LT13:   CLR     R0
2120  005320  012737  024513  000616        MOV     #MSLT13,EMADDR  ;SET TEST HEADER
2121  005326  004737  016662        LT13IT: JSR     PC,INIT1        ;GO INIT,SELECT DRIVE, SELECT ABOVE
2122  005332  012737  005410  000664        MOV     #LT13X,RTRN     ;SET RETURN ADDRESS
2123  005340  005077  173144                CLR     @C1             ;CLEAR CS1
2124  005344  005077  173216                CLR     @PSW            ;SET PRIORITY
2125  005350  052777  000100  173132        BIS     #100,@C1        ;BIT SET IE
2126  005356  005300                LT13A:  DEC     R0
2127  005360  001376                        BNE     LT13A           ;AWAIT INTERRUPT
2128  005362  012777  000340  173176  LT13E1: MOV    #340,@PSW       ;RESET PRIORITY
2129  005370  012737  022160  000666        MOV     #MSG26,ERADD    ;SET ERROR CODE
2130  005376  012737  005326  000706        MOV     #LT13IT,SCOLP   ;SET LOOP ADDRESS
2131  005404  004737  016264                JSR     PC,LTGER2       ;GO PRINT ERROR
2132  005410  004737  016534        LT13X:  JSR     PC,ITER         ;GO TO ITERATION SUBROUTINE
2133  005414  000137  002460                JMP     TSCD2           ;RETURN TO SCHED
```

```
2134
2135                                        ;THE NEXT 4 TESTS ARE MANUAL INTERVENTION STATUS TESTS.
2136                                        ;THE OPERATOR WILL BE REQUIRED TO MANIPULATE THE TE16
2137                                        ;CONTROL PANEL IN ACCORDANCE WITH TTY INSTRUCTIONS.
2138
2139                                        ;LOGIC TEST 14: STATUS AT BOT ON LINE, LOADED, NO WRITE RING**********
2140
2141  005420  032777  001000  173142  LT14:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2142  005426  001005                          BNE     LT14A           ;IF NOT: BR
2143  005430  005737  000734                  TST     STFLG           ;SEE IF SINGLE TEST
2144  005434  001433                          BEQ     LT14XX          ;IF NOT: BR
2145  005436  000137  016604                  JMP     INMT            ;ELSE GO PRINT INHIB MSG
2146  005442  012737  024560  000616  LT14A:  MOV     #MSLT14,EMADDR  ;SET TEST HEADER
2147  005450  012704  026765                  MOV     #MMSG1,R4       ;SET INSTRUCTION ONE
2148  005454  004737  017134                  JSR     PC,INST         ;GO DO INSTRUCTION
2149  005460  004737  016662          LT14IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE + SLAVE
2150  005464  012701  014602                  MOV     #14602,R1       ;SET TEST WORD
2151  005470  017702  173026                  MOV     @DS,R2          ;ASSURE MOL,WRL,DPR,DRY,BOT
2152  005474  020102                          CMP     R1,R2
2153  005476  001410                          BEQ     LT14X           ;IF SO: BR
2154  005500  012737  005460  000706          MOV     #LT14IT,SCOLP   ;SET LOOP ADDRESS
2155  005506  012737  022207  000666          MOV     #MSG27,ERADD    ;SET ERROR CODE
2156  005514  004737  016272                  JSR     PC,LTGER1       ;GO PRINT ERROR
2157  005520  004737  016534          LT14X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2158  005524  000137  002460          LT14XX: JMP     TSCD2           ;RETURN TO SCHED
2159
```

```
2160
2161                                        ;LOGIC TEST 15: STATUS AT BOT, OFFLINE, LOADED, NO WRITE RING**********
2162
2163  005530  032777  001000  173032  LT15:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2164  005536  001005                          BNE     LT15A           ;IF NOT: BR
2165  005540  005737  000734                  TST     STFLG           ;SEE IF SINGLE TEST
2166  005544  001433                          BEQ     LT15XX          ;IF NOT: BR
2167  005546  000137  016604                  JMP     INMT            ;ELSE GO PRINT INHIB MSG
2168  005552  012737  024667  000616  LT15A:  MOV     #MSLT15,EMADDR  ;SET TEST HEADER
2169  005560  012704  027063                  MOV     #MMSG2,R4
2170  005564  004737  017134                  JSR     PC,INST         ;PRINT INSTRUCTION
2171  005570  004737  016672          LT15IT: JSR     PC,INIT2        ;GO INIT, SELECT DRIVE, SLAV
2172  005574  012701  100700                  MOV     #100700,R1      ;SET TEST WORD
2173  005600  017702  172716                  MOV     @DS,R2          ;READ STATUS
2174  005604  020102                          CMP     R1,R2           ;SEE OF EXPT=RCVD
2175  005606  001410                          BEQ     LT15X
2176  005610  012737  005570  000706          MOV     #LT15IT,SCOLP   ;SET LOOP ADDRESS
2177  005616  012737  022207  000666          MOV     #MSG27,ERADD    ;SET ERROR CODE
2178  005624  004737  016272                  JSR     PC,LTGER1       ;GO PRINT ERROR
2179  005630  004737  016534          LT15X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2180  005634  000137  002460          LT15XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2181
2182                                        ;LOGIC TEST 16: STATUS AT EOT, ON-LINE, NO WRITE RING**********
2183
2184  005640  032777  001000  172722  LT16:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2185  005646  001005                          BNE     LT16A           ;IF NOT: BR
2186  005650  005737  000734                  TST     STFLG           ;SEE IF SINGLE TEST
2187  005654  001435                          BEQ     LT16XX          ;IF NOT: BR
2188  005656  000137  016604                  JMP     INMT            ;ELSE GO PRINT INHIB MSG
2189  005662  012737  024757  000616  LT16A:  MOV     #MSLT16,EMADDR  ;SET TEST HEADER
2190  005670  012704  027104                  MOV     #MMSG3,R4
2191  005674  004737  017134                  JSR     PC,INST         ;GO PRINT INSTRUCTION
2192  005700  004737  016672          LT16IT: JSR     PC,INIT2        ;SELECT DRIVE,SLAVE
2193  005704  012701  116741                  MOV     #116741,R1      ;SET TEST WORD (ATA!MOL.WRL!EOT!DPR!DRY!SSC!SLA)
2194  005710  017702  172606                  MOV     @DS,R2          ;READ STATUS
2195  005714  042702  000040                  BIC     #40,R2          ;++B MASK PES BIT
2196  005720  020102                          CMP     R1,R2           ;SEE IF EXPT=RCVD
2197  005722  001410                          BEQ     LT16X           ;IF SO: BR
2198  005724  012737  005700  000706          MOV     #LT16IT,SCOLP   ;SET LOOP ADDRESS
2199  005732  012737  022207  000666          MOV     #MSG27,ERADD    ;SET ERROR CODE
2200  005740  004737  016272                  JSR     PC,LTGER1       ;GO PRINT ERROR
2201  005744  004737  016534          LT16X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2202  005750  000137  002460          LT16XX: JMP     TSCD2           ;RETURN TO SCHED
2203
```

```
2204
2205                                        ;LOGIC TEST 17: STATUS AT ON LINE, LOADED**********
2206
2207  005754  032777  001000  172606  LT17:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2208  005762  001005                          BNE     LT17A           ;IF NOT: BR
2209  005764  005737  000734                  TST     STFLG           ;SEE IF SINGLE TEST
2210  005770  001435                          BEQ     LT17XX          ;IF NOT: BR
2211  005772  000137  016604                  JMP     INMT            ;ELSE GO PRINT INHIB MSG
2212  005776  012737  025046  000616  LT17A:  MOV     #MSLT17,EMADDR  ;SET TEST HEADER
2213  006004  012704  027142                  MOV     #MMSG4,R4
2214  006010  004737  017134                  JSR     PC,INST         ;GO PRINT INSTRUCTION
2215  006014  004737  016672          LT17IT: JSR     PC,INIT2        ;SELECT DRIVE, SLAVE
2216  006020  012701  110741                  MOV     #110741,R1      ;SET TEST WORD
2217  006024  017702  172472                  MOV     @DS,R2          ;READ STATUS
2218  006030  042702  000040                  BIC     #40,R2          ;++B MASK PES BIT
2219  006034  020102                          CMP     R1,R2           ;SEE IF EXPT=RCVD
2220  006036  001410                          BEQ     LT17X           ;IF SO: BR
2221  006040  012737  006014  000706          MOV     #LT17IT,SCOLP   ;SET LOOP ADDRESS
2222  006046  012737  022207  000666          MOV     #MSG27,ERADD    ;SET ERROR CODE
2223  006054  004737  016272                  JSR     PC,LTGER1       ;YES PRINT ERROR
2224  006060  004737  016534          LT17X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2225  006064  000137  002460          LT17XX: JMP     TSCD2           ;RETURN TO SCHED
```

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

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

006070  012737  025127  000616  LT20:   MOV     #MSLT20,EMADDR  ;SET TEST HEADER
006076  012737  006116  000706          MOV     #LT20A,SCOLP    ;SET LOOP ADDRESS
006104  012700  000022          LT20IT: MOV     #22,R0          ;SET NUMBER OF ILL CODES
006110  012737  000544  000670          MOV     #ILFT,TEMP1     ;POINT TO START IF TABLE
006116  004737  016662          LT20A:  JSR     PC,INIT1        ;GO INIT, SELECT SLAVE + DRIVE
006122  012777  177777  172362          MOV     #-1,@WC         ;SET WC= -1
006130  012701  000001                  MOV     #1,R1           ;SET TEST WORD
006134  117777  172530  172346          MOVB    @TEMP1,@C1      ;SET ILL CODE
006142  017702  172356                  MOV     @ER,R2          ;READ ER
006146  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
006150  001011                          BNE     LT20B           ;IF SO: BR
006152  012737  027445  000666          MOV     #TMS17,ERADD    ;SET ERROR CODE
006160  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
006166  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
006172  000404                          BR      LT20C
006174  020102          LT20B:  CMP     R1,R2           ;SEE UNEXPECTED ERRORS
006176  001402                  BEQ     LT20C           ;IF NOT: BR
006200  004737  015144                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
006204  005300          LT20C:  DEC     R0              ;SEE IF DONE ALL ILL CODES
006206  001403                  BEQ     LT20X           ;IF SO: BR
006210  005237  000670                  INC     TEMP1           ;BUMP ADDRESS
006214  000740                  BR      LT20A           ;CONTINUE
006216  004737  016534          LT20X:  JSR     PC,ITER         ;GO SEE IF ITERATION
006222  004737  015624                  JSR     PC,DRVCLR
006226  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2259
2260                                        ;LOGIC TEST 21: REGISTER MODIFICATION REFUSED(RMR)**********
2261
2262  006232  012737  025206  000616  LT21:   MOV     #MSLT21,EMADDR  ;SET TEST HEADER
2263  006240  012737  006246  000706          MOV     #LT21IT,SCOLP   ;SET SCOPE LOOP ADDRESS
2264  006246  004737  016662          LT21IT: JSR     PC,INIT1        ;GO INIT, SELECT SLAVE, DRIVE
2265  006252  052777  000300  172262          BIS     #300,@TC        ;SET FORMAT
2266  006260  012777  000015  172246          MOV     #15,@MR         ;SET WAM3
2267  006266  012777  000071  172214          MOV     #71,@C1         ;SET READ+GO
2268  006274  005077  172216                  CLR     @FC             ;ATTEMPT WRITE TO FC
2269  006300  012701  000004                  MOV     #4,R1           ;SET TEST WORD
2270  006304  017702  172214                  MOV     @ER,R2          ;GET ER
2271  006310  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2272  006312  001011                          BNE     LT21A           ;IF SO: BR
2273  006314  012737  027461  000666          MOV     #TMS19,ERADD    ;SET ERROR CODE
2274  006322  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
2275  006330  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2276  006334  000404                          BR      LT21B
2277  006336  020102                  LT21A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2278  006340  001402                          BEQ     LT21B           ;IF NOT: BR
2279  006342  004737  015144                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2280                                  ;++B    LT21B:  JSR     PC,ITER         ;++B DELETED GO SEE IF ITERATION
2281  006346  012703  040000          LT21B:  MOV     #40000,R3
2282  006352  005303                  LT21XA: DEC     R3              ;DELAY FOR ALPHA
2283  006354  001376                          BNE     LT21XA
2284  006356  004737  015010                  JSR     PC,EORPA        ;GO DO EOR CLEAR
2285  006362  004737  015624                  JSR     PC,DRVCLR
2286  006366  004737  016534                  JSR     PC,ITER         ;++B GO SEE IF ITERATION
2287  006372  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2288
2289                                        ;LOGIC TEST 22: CONTROL BUS PARITY (CPAR)**********
2290
2291  006376  012737  025242  000616  LT22:   MOV     #MSLT22,EMADDR  ;SET TEST HEADER
2292  006404  012737  006412  000706          MOV     #LT22IT,SCOLP   ;SET SCOPE LOOP ADDRESS
2293  006412  004737  016662          LT22IT: JSR     PC,INIT1        ;INIT, SELECT SLAVE+DRIVE
2294  006416  052777  000020  172074          BIS     #20,@CS         ;ENABLE EVEN PARITY ON MB
2295  006424  012777  177777  172064          MOV     #-1,@FC         ;WRITE TO FC
2296  006432  012701  000010                  MOV     #10,R1          ;SET TEST WORD
2297  006436  042777  000020  172054          BIC     #20,@CS         ;RESET PARITY TO ODD
2298  006444  017702  172054                  MOV     @ER,R2          ;GET ER
2299  006450  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2300  006452  001011                          BNE     LT22A           ;IF SO: BR
2301  006454  012737  027467  000666          MOV     #TMS20,ERADD    ;SET ERROR CODE
2302  006462  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
2303  006470  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2304  006474  000404                          BR      LT22X
2305  006476  020102                  LT22A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2306  006500  001402                          BEQ     LT22X           ;IF NOT: BR
2307  006502  004737  015144                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2308  006506  004737  016534          LT22X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2309  006512  004737  015624                  JSR     PC,DRVCLR
2310  006516  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2311
2312                                        ;LOGIC TEST 23: FORMAT ERROR(FMT)**********
2313
2314  006522  012737  025277  000616  LT23:   MOV     #MSLT23,EMADDR  ;SET TEST HEADER
2315  006530  012737  006536  000706          MOV     #LT23IT,SCOLP   ;SET SCOPE ADDRESS
2316  006536  004737  016662          LT23IT: JSR     PC,INIT1        ;GO INIT SELECT DRIVE+SLAVE
2317                                  ;++B    BIC     #360,@TC        ;++B DELETED SET ILLEGAL FORMAT
2318  006542  052777  000360  171772          BIS     #360,@TC        ;++B SET ILLEGAL FORMAT FOR BOTH M8906 & M8915
2319  006550  012701  000020                  MOV     #20,R1          ;SET TEST WORD
2320  006554  012777  000015  171752          MOV     #15,@MR         ;SET WAM 3
2321  006562  012777  000071  171720          MOV     #71,@C1         ;SET READ+GO
2322  006570  017702  171730                  MOV     @ER,R2          ;READ ER
2323  006574  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2324  006576  001011                          BNE     LT23A           ;IF SO: BR
2325  006600  012737  027476  000666          MOV     #TMS21,ERADD    ;SET ERROR CODE
2326  006606  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
2327  006614  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2328  006620  000404                          BR      LT23X
2329  006622  020102          LT23A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2330  006624  001402                          BEQ     LT23X           ;IF NOT: BR
2331  006626  004737  015144                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2332  006632  004737  016534          LT23X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2333  006636  004737  015010                  JSR     PC,EORPA
2334  006642  004737  015624                  JSR     PC,DRVCLR
2335  006646  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2336                                        ;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)**********
2337
2338  006652  012737  025344  000616  LT24:   MOV     #MSLT24,EMADDR  ;SET TEST HEADER
2339  006660  012737  006666  000706          MOV     #LT24IT,SCOLP   ;SET SCOPE ADDRESS
2340  006666  012737  000005  000602  LT24IT: MOV     #5,ITAMT
2341  006674  004737  016710                  JSR     PC,INIT3        ;GO INIT, SELECT DRIVE+SLAVE
2342  006700  052777  000300  171634          BIS     #300,@TC        ;SET NORMAL FORMAT
2343  006706  012777  027776  171600          MOV     #WDATA,@BA      ;SET BA
2344  006714  012777  177760  171574          MOV     #-20,@FC        ;SET FC
2345  006722  012777  177770  171562          MOV     #-10,@WC        ;SET WC
2346  006730  012777  000013  171576          MOV     #13,@MR         ;SELECT WAM 2
2347  006736  012777  000061  171544          MOV     #61,@C1         ;SET WRITE+GO
2348  006744  052777  000020  171546          BIS     #20,@CS         ;FORCE EVEN PARITY
2349  006752  012701  000040                  MOV     #40,R1          ;SET TEST WORD
2350  006756  012703  000004                  MOV     #4,R3
2351  006762  005000                          CLR     R0
2352  006764  005300                  1$:     DEC     R0
2353  006766  001376                          BNE     1$              ;DELAY
2354  006770  005303                          DEC     R3
2355  006772  001374                          BNE     1$
2356  006774  012700  000004                  MOV     #4,R0
2357  007000  012777  000013  171526  LT24B:  MOV     #13,@MR         ;CLOCK MR 4 TIMES
2358  007006  005300                          DEC     R0
2359  007010  022700  000002                  CMP     #2,R0           ;SEE IF DONE 1 BYTE
2360  007014  001002                          BNE     LT24B0          ;IF NOT: BR
2361  007016  017701  171512                  MOV     @MR,R1          ;ELSE GET BYTE 1
2362  007022  005700                  LT24B0: TST     R0              ;SEE IF BYTE 2
2363  007024  001365                          BNE     LT24B           ;IF NOT: BR
2364  007026  017704  171502                  MOV     @MR,R4          ;GET BYTE 2
2365  007032  005000                          CLR     R0
2366  007034  005300                  LT24C:  DEC     R0
2367  007036  001376                          BNE     LT24C           ;DELAY
2368  007040  032777  000040  171456          BIT     #40,@ER         ;SEE IF DPAR IS SET
2369  007046  001023                          BNE     LT24D           ;IF SO: BR
2370  007050  000301                          SWAB    R1
2371  007052  042701  177400                  BIC     #177400,R1      ;GET LOW BYTE
2372  007056  042704  000377                  BIC     #377,R4
2373  007062  050401                          BIS     R4,R1           ;GET HIGH BYTE
2374  007064  005237  000740                  INC     T24FL           ;SET T24 FLAG
2375  007070  012737  027504  000666          MOV     #TMS22,ERADD    ;SET ERROR CODE
2376  007076  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
2377  007104  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2378  007110  005037  000740                  CLR     T24FL           ;CLEAR FLAG
2379  007114  000412                          BR      LT24X
2380  007116  012701  000050          LT24D:  MOV     #50,R1
2381  007122  017702  171376                  MOV     @ER,R2          ;GET ERROR REGISTER
2382  007126  042702  020000                  BIC     #20000,R2       ;MASK OPI
2383  007132  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2384  007134  001402                          BEQ     LT24X           ;IF NOT: BR
2385  007136  004737  015144                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2386  007142  042777  000020  171350  LT24X:  BIC     #20,@CS         ;RESET EVEN PARITY
2387  007150  004737  015010                  JSR     PC,EORPA        ;GO DO EOR CLEAR
2388  007154  004737  015624                  JSR     PC,DRVCLR       ;GO SEE IF DRIVE CLEAR OK
2389  007160  004737  016534                  JSR     PC,ITER         ;GO SEE IF ITERATION
2390  007164  012737  000020  000602          MOV     #20,ITAMT
2391  007172  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2392
2393                                        ;LOGIC TEST 25: NON-EXECUTABLE FUNCTION(NEF)**********
2394
2395  007176  012737  025404  000616  LT25:   MOV     #MSLT25,EMADDR  ;SET TEST HEADER
2396  007204  004737  016710          LT25IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2397  007210  052777  000300  171324          BIS     #300,@TC        ;SET NORMAL FORMAT
2398  007216  012777  177777  171272          MOV     #-1,@FC         ;SET ITLLEGAL FC
2399  007224  012777  000013  171302          MOV     #13,@MR         ;SET WAM 2
2400  007232  012777  000061  171250          MOV     #61,@C1         ;LOAD WRITE+GO
2401  007240  012701  004000                  MOV     #4000,R1        ;SET TEST WORD
2402  007244  017702  171254                  MOV     @ER,R2          ;GET ER
2403  007250  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2404  007252  001014                          BNE     LT25A           ;IF SO: BR
2405  007254  012737  007204  000706          MOV     #LT25IT,SCOLP   ;SET LOOP ADDRESS
2406  007262  012737  027572  000666          MOV     #TMS31,ERADD    ;SET ERROR CODE
2407  007270  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLAG
2408  007276  004737  015156                  JSR     PC,LTGERO       ;GO PRINT ERROR
2409  007302  000404                          BR      LT25X
2410  007304  020102                  LT25A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2411  007306  001402                          BEQ     LT25X           ;IF NOT: BR
2412  007310  004737  015144                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2413  007314  004737  016534          LT25X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2414  007320  004737  015624                  JSR     PC,DRVCLR
2415  007324  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2416
2417                                        ;LOGIC TEST 26: FRAME COUNT ERROR(FCE)**********
2418
2419  007330  012737  025440  000616  LT26:   MOV     #MSLT26,EMADDR  ;SET TEST HEADER
2420  007336  004737  016710          LT26IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2421  007342  005000                          CLR     R0
2422  007344  005300                  1$:     DEC     R0
2423  007346  001376                          BNE     1$              ;AWAIT OPI RESET
2424  007350  052777  000300  171164          BIS     #300,@TC        ;SET NORMAL FORMAT
2425  007356  012777  177770  171126          MOV     #-10,@WC        ;SET WC=-10
2426  007364  012777  177760  171124          MOV     #-20,@FC        ;SET FC=-20
2427  007372  012777  000013  171134          MOV     #13,@MR         ;SET WAM 3
2428  007400  012777  000061  171102          MOV     #61,@C1         ;LOAD WRITE+GO
2429  007406  012701  001000                  MOV     #1000,R1        ;SET TEST WORD
2430  007412  005000                          CLR     R0
2431  007414  005300                  2$:     DEC     R0
2432  007416  001376                          BNE     2$              ;DELAY
2433  007420  012777  000025  171106          MOV     #25,@MR         ;LOAD MM EOR CLEAR
2434  007426  105077  171102                  CLRB    @MR             ;RESET MR
2435  007432  012703  000004                  MOV     #4,R3
2436  007436  005000                          CLR     R0
2437  007440  032777  001000  171056  3$:     BIT     #1000,@ER       ;SEE IF FCE SET
2438  007446  001022                          BNE     4$              ;IF SO: BR
2439  007450  005300                          DEC     R0
2440  007452  001372                          BNE     3$              ;DELAY
2441  007454  005303                          DEC     R3
2442  007456  001370                          BNE     3$
2443  007460  017702  171040                  MOV     @ER,R2          ;GET ER
2444  007464  012737  007336  000706          MOV     #LT26IT,SCOLP   ;SET SCOPE ADDRESS
2445  007472  012737  027551  000666          MOV     #TMS28,ERADD
2446  007500  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
2447  007506  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2448  007512  000406                          BR      LT26X
2449  007514  017702  171004          4$:     MOV     @ER,R2          ;GET ERROR REGISTER
2450  007520  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2451  007522  001402                          BEQ     LT26X           ;IF NOT: BR
2452  007524  004737  015144                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2453  007530  004737  016534  LT26X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2454  007534  004737  015624                  JSR     PC,DRVCLR
2455  007540  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2456
2457                                        ;LOGIC TEST 27: ILLEGAL REGISTER(ILR)**********
2458
2459  007544  022737  172400  000510  LT27:   CMP     #172400,C1      ;SEE IF ADDRESSES OPEN
2460  007552  001041                          BNE     LT27XX          ;IF NOT: BR
2461  007554  012737  007600  000706          MOV     #LT27A,SCOLP    ;SET SCOPE ADDRESS
2462  007562  012737  025474  000616          MOV     #MSLT27,EMADDR  ;SET TEST HEADER
2463  007570  012700  000020          LT27IT: MOV     #20,R0          ;SET NUMBER OF ILR TESTS
2464  007574  012701  172434                  MOV     #172434,R1      ;SET FIRST ILR ADDRESS
2465  007600  004737  016710          LT27A:  JSR     PC,INIT3        ;GO INIT, SELECT DRIVE+SLAVE
2466  007604  011103                          MOV     (R1),R3         ;ATTEMPT ILR READ
2467  007606  032777  000002  170710          BIT     #2,@ER          ;SEE IF ILR=1
2468  007614  001010                          BNE     LT27B           ;IF SO: BR
2469  007616  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT-NOT RCVD FLAG
2470  007624  012737  027373  000666          MOV     #TMS10,ERADD    ;SET ERROR CODE
2471  007632  004737  015164                  JSR     PC,LTGER        ;GO PRINT ERROR
2472  007636  005300                  LT27B:  DEC     R0              ;SEE IF DONE ALL
2473  007640  001402                          BEQ     LT27X           ;IF SO: BR
2474  007642  005721                          TST     (R1)+           ;BUMP ADDRESS
2475  007644  000755                          BR      LT27A           ;CONTINUE TESTS
2476  007646  004737  016534          LT27X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2477  007652  004737  015624                  JSR     PC,DRVCLR
2478  007656  000137  002460          LT27XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2479
2480                                        ;LOGIC TEST 30: DRIVE TIMING ERROR**********
2481
2482  007662  012737  027600  000666  LT30:   MOV     #TMS32,ERADD    ;SET ERROR CODE
2483  007670  012737  025530  000616          MOV     #MSLT30,EMADDR  ;SET TEST HEADER
2484  007676  012737  007704  000706          MOV     #LT30IT,SCOLP   ;SET SCOPE ADDRESS
2485  007704  004737  016710          LT30IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE + SLAVE
2486  007710  052777  000300  170624          BIS     #300,@TC        ;SET NORMAL FORMAT
2487  007716  012701  010000                  MOV     #10000,R1       ;SET TEST WORD
2488  007722  012777  000017  170604          MOV     #17,@MR         ;CRIPPLE OCCUPIED
2489  007730  005077  170562                  CLR     @FC             ;SET FC3
2490  007734  012777  000061  170546          MOV     #61,@C1         ;LOAD WRITE+GO
2491  007742  032777  010000  170554          BIT     #10000,@ER      ;SEE IF DTE SET
2492  007750  001005                          BNE     LT30A           ;IF SO: BR
2493  007752  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
2494  007760  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2495  007764  004737  016710          LT30A:  JSR     PC,INIT3        ;GO INIT SELECT DRIVE,SLAVE
2496  007770  052777  000300  170544          BIS     #300,@TC        ;SET FORMAT
2497  007776  012701  010000                  MOV     #10000,R1       ;SET TEST WORD
2498  010002  005077  170510                  CLR     @FC             ;SET FCS
2499  010006  012777  000015  170520          MOV     #15,@MR         ;SET WRAP 3
2500  010014  012777  000061  170466          MOV     #61,@C1         ;LOAD WRITE+GO
2501  010022  012704  040000                  MOV     #40000,R4
2502  010026  005777  170510          LT30B:  TST     @TC             ;SEE IF ALPHA
2503  010032  100015                          BPL     LT30C           ;AWAIT ALPHA
2504  010034  005300                          DEC     R0
2505  010036  001373                          BNE     LT30B
2506  010040  013704  000616                  MOV     EMADDR,R4
2507  010044  004737  017662                  JSR     PC,TTOUT        ;PRINT HEADER
2508  010050  012704  022743                  MOV     #MSG50,R4
2509  010054  004737  017662                  JSR     PC,TTOUT        ;PRINT ALPHA ERROR
2510  010060  004737  016504                  JSR     PC,SCOPE
2511  010064  000435                          BR      LT30X
2512  010066  012777  000015  170440  LT30C:  MOV     #15,@MR         ;CLOCK MR
2513  010074  012777  000015  170432          MOV     #15,@MR         ;CLOCK MR
2514  010102  005000                          CLR     R0
2515  010104  005300                  LT30D:  DEC     R0
2516  010106  001376                          BNE     LT30D           ;DELAY
2517  010110  032777  010000  170406          BIT     #10000,@ER      ;SEE IF DTE SET
2518  010116  001006                          BNE     LT30E           ;IF SO: BR
2519  010120  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
2520  010126  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2521  010132  000412                          BR      LT30X
2522  010134  012701  010000          LT30E:  MOV     #10000,R1       ;SET TEST WORD
2523  010140  017702  170360                  MOV     @ER,R2          ;GET ERROR REGISTER
2524  010144  042702  020100                  BIC     #20100,R2       ;MASK OPI AND VPE
2525  010150  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2526  010152  001402                          BEQ     LT30X           ;IF NOT: BR
2527  010154  004737  015144                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2528  010160  004737  016534          LT30X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2529  010164  004737  015010                  JSR     PC,EORPA        ;GO CLEAR GO BIT
2530  010170  004737  015624                  JSR     PC,DRVCLR
2531  010174  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
2532
```

```
2533
2534                                        ;LOGIC TEST 31: OPERATION INCOMPLETE(OPI)**********
2535
2536  010200  012737  025566  000616  LT31:   MOV     #MSLT31,EMADDR  ;SET TEST HEADER
2537  010206  012737  010206  000706  LT31IT: MOV     #LT31IT,SCOLP   ;SET SCOPE ADDRESS
2538  010214  012737  027614  000666          MOV     #TMS33A,ERADD   ;SET ERROR MSG HDR
2539  010222  012737  000002  000602          MOV     #2,ITAMT        ;SET REDUCED ITER COUNT
2540  010230  004737  016710                  JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2541  010234  005000                          CLR     R0
2542  010236  005300                  1$:     DEC     R0
2543  010240  001376                          BNE     1$              ;AWAIT OPI RESET
2544  010242  052777  000300  170272          BIS     #300,@TC        ;SET FORMAT
2545  010250  012777  000013  170256          MOV     #13,@MR         ;SET WAM 2
2546  010256  005077  170234                  CLR     @FC             ;SET FRAME COUNT
2547  010262  012705  020000                  MOV     #20000,R5       ;SET TEST BIT (OPI)
2548  010266  012702  010304                  MOV     #2$,R2          ;SET RETURN ADDRESS FROM TIMER
2549  010272  004737  010504                  JSR     PC,TIMON        ;START TIMER
2550  010276  012777  000061  170204          MOV     #61,@C1         ;LOAD WRITE+GO
2551  010304  030577  170214          2$:     BIT     R5,@ER          ;BRANCH WHEN OPI SETS
2552  010310  001002                          BNE     3$
2553  010312  000163  010576                  JMP     TIMER(R3)       ;GO TO TIMER & RETURN TO 2$ ABOVE
2554  010316  017702  170202          3$:     MOV     @ER,R2          ;GET ERROR REGISTER
2555  010322  020502                          CMP     R5,R2           ;SEE IF UNEXPECTED ERRORS
2556  010324  001403                          BEQ     4$              ;IF NOT: BR
2557  010326  004737  015144                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2558  010332  000453                          BR      LT31X
2559  010334  004737  010670          4$:     JSR     PC,TIMOK        ;GO CHECK TIME FOR OPI TO SET
2560  010340  102450                          BVS     LT31X           ;BRANCH IF TIME WAS INCORRECT
2561
2562  010342  012737  010356  000706          MOV     #LT31A,SCOLP    ;SET SCOPE LOOP
2563  010350  012737  027630  000666          MOV     #TMS33B,ERADD   ;SET ERROR MSG HEADER
2564  010356  004737  016710          LT31A:  JSR     PC,INIT3        ;GO INIT
2565  010362  005000                          CLR     R0
2566  010364  005300                  1$:     DEC     R0              ;WAIT FOR OPI TO CLEAR
2567  010366  001376                          BNE     1$
2568  010370  052777  000300  170144          BIS     #300,@TC        ;SET FORMAT
2569  010376  012777  000015  170130          MOV     #15,@MR         ;SET WRAP 3
2570  010404  012702  010426                  MOV     #2$,R2          ;SET RETURN ADDRESS FROM TIMER
2571  010410  012705  020000                  MOV     #20000,R5       ;SET TEST WORD
2572  010414  004737  010504                  JSR     PC,TIMON        ;START TIMER
2573  010420  012777  000071  170062          MOV     #71,@C1         ;LOAD READ+GO
2574  010426  030577  170072          2$:     BIT     R5,@ER          ;BRANCH WHEN OPI SETS
2575  010432  001002                          BNE     3$
2576  010434  000163  010576                  JMP     TIMER(R3)       ;GO TO TIMER
2577  010440  017702  170060          3$:     MOV     @ER,R2          ;GET ERROR REGISTER
2578  010444  020502                          CMP     R5,R2           ;SEE IF UNEXPECTED ERRORS
2579  010446  001403                          BEQ     4$
2580  010450  004737  015144                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2581  010454  000402                          BR      LT31X           ;EXIT TEST
2582  010456  004737  010670          4$:     JSR     PC,TIMOK        ;GO CHECK TIME
2583  010462  004737  016534          LT31X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2584  010466  004737  015624                  JSR     PC,DRVCLR
2585  010472  012737  000020  000602          MOV     #20,ITAMT
2586  010500  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
2587
2588                                        ;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.
010504  005000                  TIMON:  CLR     R0              ;CLEAR TICK COUNT
010506  005001                          CLR     R1
010510  012703  000024                  MOV     #24,R3          ;PRESET INDEX TO TIMER
010514  032777  000100  170012          BIT     #100,@MR        ;BRANCH IF OSC CLEAR
010522  001405                          BEQ     2$
010524  032777  000100  170002  1$:     BIT     #100,@MR        ;WAIT FOR OSC TO CLEAR
010532  001374                          BNE     1$
010534  000405                          BR      4$              ;EXIT

010536  005403                  2$:     NEG     R3              ;SET INDEX TO TIMER
010540  032777  000100  167766  3$:     BIT     #100,@MR        ;WAIT FOR OSC TO SET
010546  001774                          BEQ     3$
010550  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
                                        ;IHE OSC BIT CHANGES STATE R3 IS NEGATED.
010552  032777  000100  167754  TIMER1: BIT     #100,@MR        ;BRANCH IF OSC HAS CHANGED STATE
010560  001406                          BEQ     TIMER
010562  000112                          JMP     (R2)            ;RETURN
        010576                          .=TIMER1+24
010576  005403                  TIMER:  NEG     R3              ;SET INDEX TO OTHER STATE
010600  062700  000001                  ADD     #1,R0           ;INCREMENT TICK COUNT
010604  005501                          ADC     R1
010606  022701  000003                  CMP     #3,R1           ;BRANCH IF TIMER OVERFLOWS
010612  001410                          BEQ     TIMOVF
010614  000112                          JMP     (R2)            ;RETURN
        010622                          .-TIMER +24
010622  032777  000100  167704  TIMER0: BIT     #100,@MR        ;BRANCH IF OSC SET
010630  001362                          BNE     TIMER
010632  000112                          JMP     (R2)            ;RETURN

010634  013704  000616          TIMOVF: MOV     EMADDR,R4       ;TYPE TEST HEADER
010640  004737  017662                  JSR     PC,TTOUT
010644  013704  000666                  MOV     ERADD,R4        ;GET ERROR MSG ADDRESS
010650  004737  017662                  JSR     PC,TTOUT        ;AND TYPE IT
010654  012704  027710                  MOV     #TMS33E,R4      ;TYPE
010660  004737  017662                  JSR     PC,TTOUT        ;'TIMER OVERFLOWED'
010664  000137  010462                  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 9,500,000 USECS (9.5 SECS). THE 448 IS DERIVED FROM
                                ;56 USECS/TICK TIMES THE DIVISION BY 8 BY THE TIMOK ROUTINE.
010670  000240                  TIMOK:  NOP
010672  006201                          ASR     R1              ;DIVIDE COUNT BY 8
010674  006000                          ROR     R0
010676  006201                          ASR     R1
010700  006000                          ROR     R0
010702  006201                          ASR     R1
010704  006000                          ROR     R0
```

```
2645  010706  013701  001010                MOV     SLVTYP,R1       ;++B GET SLAVE TYPE (0/1 = TE16/TU77)
2646  010712  006301                        ASL     R1              ;++B FORM INDEX
2647  010714  020061  011020                CMP     R0,200$(R1)     ;++B BRANCH IF GREATER THAN LOWER LIMIT(5.5 SECS)
2648  010720  101016                        BHI     1$
2649  010722  013704  000616                MOV     EMADDR,R4       ;GET ERROR MSG HEADER
2650  010726  004737  017662                JSR     PC,TTOUT        ;TYPE ERROR MSG HEADER
2651  010732  013704  000666                MOV     ERADD,R4        ;GET ERROR DESCRIPTOR MSG
2652  010736  004737  017662                JSR     PC,TTOUT
2653  010742  012704  027643                MOV     #TMS33C,R4      ;TYPE 'OCCURED TOO SOON'
2654  010746  004737  017662                JSR     PC,TTOUT
2655  010752  000262                        SEV                     ;SET 'V' TO INDICATE ERROR
2656  010754  000420                        BR      2$
2657
2658  010756  020061  011024        1$:     CMP     R0,201$(R1)     ;++B BRANCH IF LESS THAN UPPER LIMIT(9.5 SECS)
2659  010762  003415                        BLE     2$
2660  010764  013704  000616                MOV     EMADDR,R4       ;GET ERROR MSG HEADER
2661  010770  004737  017662                JSR     PC,TTOUT
2662  010774  013704  000666                MOV     ERADD,R4
2663  011000  004737  017662                JSR     PC,TTOUT        ;TYPE ERROR MSG HEADER
2664  011004  012704  027665                MOV     #TMS33D,R4      ;TYPE 'OCCURED TOO LATE'
2665  011010  004737  017662                JSR     PC,TTOUT
2666  011014  000262                        SEV
2667  011016  000207                2$:     RTS     PC
2668
2669
2670                                  ;++B TABLE OF MIN AND MAX TIMES FOR OPI FOR TE16 AND TU77 SLAVES
2671                                  ;++B MIN TIMES (5.5 SECS)
2672  011020  027764                200$:   .WORD   12276.          ;++B TE16
2673  011022  020622                        .WORD   8594.           ;++B TU77
2674
2675                                  ;++B MAX TIMES (9.5 SECS)
2676  011024  051325                201$:   .WORD   21205.          ;++B TE16
2677  011026  034774                        .WORD   14844.          ;++B TU77
```

```
2678
2679
2680
2681                                        ;LOGIC TEST 32: UNSAFE(UNS)**********
2682
2683  011030  012737  025622  000616  LT32:   MOV     #MSLT32,EMADDR  ;SET TEST HEADER
2684  011036  012737  011044  000706          MOV     #LT32IT,SCOLP   ;SET SCOPE ADDRESS
2685  011044  004737  016710          LT32IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE +SLAVE
2686  011050  013700  000660                  MOV     SLVN,R0         ;GET SLAVE NUMBER
2687  011054  012701  040000                  MOV     #40000,R1       ;++B SET TEST WORD (UNS)
2688  011060  032777  000004  167450          BIT     #4,@DT          ;++B BRANCH IF TE16
2689  011066  001402                          BEQ     1$              ;++B
2690  011070  052701  004000                  BIS     #4000,R1        ;++B SET ALSO NEF FOR TU77
2691  011074  005100                  1$:     COM     R0              ;SET NONEXISTANT SLAVE
2692  011076  042700  177770                  BIC     #177770,R0      ;MASK SLAVE NUMBER
2693  011102  052700  000300                  BIS     #300,R0         ;SET FORMAT
2694  011106  010077  167430                  MOV     R0,@TC          ;SELECT ILLEGAL SLAVE
2695  011112  032777  002000  167416          BIT     #2000,@DT       ;EXIT TEST IF SALVE AVAILABLE
2696  011120  001030                          BNE     LT32XX
2697  011122  012777  000071  167360          MOV     #71,@C1         ;LOAD READ+GO
2698  011130  017702  167370                  MOV     @ER,R2          ;READ ER
2699  011134  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2700  011136  001011                          BNE     2$              ;IF SO: BR
2701  011140  012737  027730  000666          MOV     #TMS34,ERADD    ;SET ERROR CODE
2702  011146  012737  000001  000712          MOV     #1,EXFL         ;SET ERROR CODE
2703  011154  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2704  011160  000404                          BR      LT32X
2705  011162  020102                  2$:     CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2706  011164  001402                          BEQ     LT32X           ;IF NOT: BR
2707  011166  004737  015144                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2708  011172  004737  016534          LT32X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2709  011176  004737  015624                  JSR     PC,DRVCLR
2710  011202  000137  002460          LT32XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2711
2712                                        ;THE FOLLOWING 6 TESTS WILL LOOK AT VARIOUS BITS IN THE
2713                                        ;DRIVE STATUS(DS) AND TAPE CONTROL(TC)
2714                                        ;REGISTERS BY FORCING CERTAIN CONDITONS WHICH DO NOT
2715                                        ;REQUIRE TAPE MOVEMENT.
2716
2717                                        ;LOGIC TEST 33: POSITIONING IN PROGRESS(PIP)**********
2718
2719  011206  012737  025656  000616  LT33:   MOV     #MSLT33,EMADDR  ;SET TEST HEADER
2720  011214  012737  011222  000706          MOV     #LT33IT,SCOLP   ;SET SCOPE ADDRESS
2721  011222  004737  016710          LT33IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2722  011226  012777  000013  167300          MOV     #13,@MR         ;SET WAM 2
2723  011234  012777  177777  167254          MOV     #-1,@FC         ;SET FCS
2724  011242  012777  000031  167240          MOV     #31,@C1         ;LOAD SPACE FORWARD+GO
2725  011250  032777  020000  167244          BIT     #20000,@DS      ;SEE IF PIP=1
2726  011256  001010                          BNE     LT33X           ;IF SO: BR
2727  011260  012737  027423  000666          MOV     #TMS14,ERADD    ;SET ERROR CODE
2728  011266  012737  000001  000712          MOV     #1,EXFL         ;SET ERROR CODE
2729  011274  004737  015156                  JSR     PC,LTGERO       ;GO PRINT ERROR
2730  011300  004737  016534          LT33X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2731  011304  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2732
2733                                        ;LOGIC TEST 34: PHASE ENCODED STATUS(PES)********
2734
2735  011310  012737  027343  000666  LT34:   MOV     #TMS6,ERADD     ;SET ERROR CODE
2736  011316  012737  025712  000616          MOV     #MSLT34,EMADDR  ;SET TEST HEADER
2737  011324  012700  000004          LT34IT: MOV     #4,R0
2738  011330  004737  016710          LT34A1: JSR     PC,INIT3        ;GO INIT, SELECT DRIVE+SLAVE
2739  011334  042777  003400  167200          BIC     #3400,@TC       ;SELECT NRZI
2740  011342  052777  001400  167172          BIS     #1400,@TC
2741  011350  032777  000040  167144  LT34A:  BIT     #40,@DS         ;SEE IF PES=0
2742  011356  001410                          BEQ     LT34B           ;IF SO: BR
2743  011360  012737  000002  000712          MOV     #2,EXFL         ;SET RCVD-NOT EXPT
2744  011366  012737  011330  000706          MOV     #LT34A1,SCOLP   ;SET SCOPE ADDRESS
2745  011374  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2746  011400  004737  016724          LT34B:  JSR     PC,INIT4
2747  011404  032777  000040  167110  LT34C:  BIT     #40,@DS         ;SEE IF PES=1
2748  011412  001010                          BNE     LT34X           ;IF SO: BR
2749  011414  012737  011404  000706          MOV     #LT34C,SCOLP    ;SET SCOPE ADDRESS
2750  011422  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT-NOT RCVD FLAG
2751  011430  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2752  011434  004737  016534          LT34X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2753  011440  000137  002460          LT34XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2754
2755                                        ;LOGIC TEST 35: SLAVE ADDRESS CHANGE (SAC)**********
2756
2757  011444  012737  027753  000666  LT35:   MOV     #TMS37,ERADD
2758  011452  012737  025746  000616          MOV     #MSLT35,EMADDR
2759  011460  004737  016710          LT35IT: JSR     PC,INIT3        ;INIT SELECT DRIVE, SLAVE
2760  011464  032777  000020  167030  1$:     BIT     #20,@DS         ;SEE IF SDWN IS RESET
2761  011472  001374                          BNE     1$              ;IF NOT: BR
2762  011474  052777  000300  167040          BIS     #300,@TC        ;SET FORMAT
2763  011502  012777  000015  167024          MOV     #15,@MR         ;SET WAM 3
2764  011510  012777  000071  166772          MOV     #71,@C1         ;LOAD READ+GO
2765  011516  032777  020000  167016          BIT     #20000,@TC      ;SEE IF SAC=0
2766  011524  001410                          BEQ     LT35A           ;IF SO: BR
2767  011526  012737  000002  000712          MOV     #2,EXFL         ;SET RCV-NOT EXPT FLAG
2768  011534  012737  011460  000706          MOV     #LT35IT,SCOLP   ;SET SCOPE ADDRESS
2769  011542  004737  015156                  JSR     PC,LTGERO       ;GO PRINT ERROR
2770  011546  004737  016710          LT35A:  JSR     PC,INIT3        ;INIT
2771  011552  005277  166764                  INC     @TC             ;BUMP SLAVE ADDRESS
2772  011556  032777  020000  166756          BIT     #20000,@TC      ;SEE IF SAC=1
2773  011564  001010                          BNE     LT35X           ;IF SO: BR
2774  011566  012737  011546  000706          MOV     #LT35A,SCO_P    ;SET SCOPE ADDRESS
2775  011574  012737  000001  000712          MOV     #1,EXFL         ;SE EXPT-NOT RCVD FLAG
2776  011602  004737  015156                  JSR     PC,LTGERO       ;GO PRINT ERROR
2777  011606  004737  016534          LT35X:  JSR     PC,ITER
2778  011612  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2779
2780                                        ;LOGIC TEST 36: FRAME COUNTER STATUS(FCS)**********
2781
2782  011616  012737  026013  000616  LT36:   MOV     #MSLT36,EMADDR
2783  011624  012737  027761  000666          MOV     #TMS38,ERADD    ;SET ERROR CODE
2784  011632  004737  016710          LT36IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2785  011636  032777  040000  166676          BIT     #40000,@TC      ;SEE IF FCS=0
2786  011644  001410                          BEQ     1$              ;IF SO: BR
2787  011646  012737  011632  000706          MOV     #LT36IT,SCOLP   ;SET SCOPE ADDRESS
2788  011654  012737  000002  000712          MOV     #2,EXFL         ;SET RCVD-NOT EXPT
2789  011662  004737  015156                  JSR     PC,LTGERO       ;GO PRINT ERROR
2790  011666  004737  016710          1$:     JSR     PC,INIT3        ;INIT
2791  011672  005077  166620                  CLR     @FC             ;WRITE TO FC
2792  011676  032777  040000  166636          BIT     #40000,@TC      ;SEE IF FCS=1
2793  011704  001010                          BNE     LT36X           ;IF SO: BR
2794  011706  012737  011666  000706          MOV     #1$,SCOLP       ;SET SCOPE ADDRESS
2795  011714  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT-NOT RCVD
2796  011722  004737  015156                  JSR     PC,LTGERO       ;GO PRINT ERROR
2797  011726  004737  016534          LT36X:  JSR     PC,ITER
2798  011732  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2799
2800                                                    ;LOGIC TEST 37: ACCELERATION(ACCL)**********
2801
2802  011736  012737  026060  000616  LT37:   MOV     #MSLT37,EMADDR
2803  011744  012737  027767  000666          MOV     #TMS39,ERADD    ;SET ERROR CODE
2804  011752  004737  016710          LT37IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2805  011756  052777  000300  166556          BIS     #300,@TC        ;SET FORMAT
2806  011764  005777  166552                  TST     @TC             ;SEE IF ACCL=1
2807  011770  100410                          BMI     LT37A           ;IF SO: BR
2808  011772  012737  000001  000712          MOV     #1,EXFL
2809  012000  012737  011752  000706          MOV     #LT37IT,SCOLP   ;SET SCOPE ADDRESS
2810  012006  004737  015156                  JSR     PC,LTGERO       ;GO PRINT ERROR
2811  012012  004737  016710          LT37A:  JSR     PC,INIT3        ;INIT
2812  012016  052777  000300  166516          BIS     #300,@TC        ;SET FORMAT
2813  012024  012777  000015  166502          MOV     #15,@MR         ;SET WAM 3
2814  012032  012777  000071  166450          MOV     #71,@C1         ;LOAD READ+GO
2815  012040  012700  100000                  MOV     #100000,R0      ;SET ACCL DELAY
2816  012044  005777  166472          LT37B:  TST     @TC             ;SEE IF ACCL=0
2817  012050  100012                          BPL     LT37X           ;IF SO: BR
2818  012052  005300                          DEC     R0
2819  012054  001373                          BNE     LT37B           ;DELAY
2820  012056  012737  012012  000706          MOV     #LT37A,SCOLP    ;SET SCOPE ADDRESS
2821  012064  012737  000002  000712          MOV     #2,EXFL
2822  012072  004737  015156                  JSR     PC,LTGERO       ;GO PRINT ERROR
2823  012076  004737  016534          LT37X:  JSR     PC,ITER
2824  012102  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2825
2826                                        ;LOGIC TEST 40: PE TAPE MARK (TM)**********
2827
2828  012106  012737  012122  000706  LT40:   MOV     #LT40IT,SCOLP   ;SET SCOPE ADDRESS
2829  012114  012737  026126  000616          MOV     #MSLT40,EMADDR
2830  012122  004737  016724          LT40IT: JSR     PC,INIT4        ;INIT, SELECT DRIVE+SLAVE
2831  012126  005000                  1$:     CLR     R0
2832  012130  005300                          DEC     R0
2833  012132  001376                          BNE     1$              ;DELAY FOR OPI RESET
2834  012134  052777  002300  166400          BIS     #2300,@TC
2835  012142  012777  000007  166364          MOV     #7,@MR          ;SET WAM 0
2836  012150  012777  000027  166332          MOV     #27,@C1         ;LOAD WRITE TAPE MARK+GO
2837  012156  012700  100000                  MOV     #100000,R0      ;SET DELAY
2838  012162  032777  000004  166332  2$:     BIT     #4,@DS          ;SEE IF TM=1
2839  012170  001012                          BNE     LT40X           ;IF SO: BR
2840  012172  005300                          DEC     R0
2841  012174  001372                          BNE     2$              ;DELAY
2842  012176  012737  027321  000666          MOV     #TMS3,ERADD
2843  012204  012737  000001  000712          MOV     #1,EXFL
2844  012212  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2845  012216  004737  016534          L 40X:  JSR     PC,ITER
2846  012222  000137  002460          LT40XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2847
2848                                                ;LOGIC TEST 41: NRZ TAPE MARK (TM,VPE,ITM)**********
2849
2850  012226  012737  012242  000706  LT41:   MOV     #LT41IT,SCOLP   ;SET SCOPE ADDRESS
2851  012234  012737  026173  000616          MOV     #MSLT41,EMADDR
2852  012242  004737  016710          LT41IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE,SLAVE
2853  012246  052777  001700  166266          BIS     #1700,@TC       ;SET NRZ+NORMAL FORMAT
2854  012254  012777  177760  166234          MOV     #-20,@FC        ;SET FCS
2855  012262  012777  000007  166244          MOV     #7,@MR          ;SET WAM 0
2856  012270  012777  000027  166212          MOV     #27,@C1         ;LOAD WRITE TAPE MARK+GO
2857  012276  005000                          CLR     R0
2858  012300  032777  000004  166214  1$:     BIT     #4,@DS          ;SEE IF TM=1
2859  012306  001012                          BNE     2$              ;IF SO: BR
2860  012310  005300                          DEC     R0
2861  012312  001372                          BNE     1$              ;DELAY
2862  012314  012737  027321  000666          MOV     #TMS3,ERADD     ;SET ERROR CODE
2863  012322  012737  000001  000712          MOV     #1,EXFL
2864  012330  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2865  012334  032777  002000  166162  2$:     BIT     #2000,@ER       ;SEE IF ITM=1
2866  012342  001010                          BNE     3$              ;IF SO: BR
2867  012344  012737  027564  000666          MOV     #TMS30,ERADD    ;SET ERROR CODE
2868  012352  012737  000001  000712          MOV     #1,EXFL
2869  012360  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2870  012364  032777  000100  166132  3$:     BIT     #100,@ER        ;SEE IF VPE-1
2871  012372  001011                          BNE     4$              ;IF SO: BR
2872  012374  012737  027551  000666          MOV     #TMS28,ERADD    ;SET ERROR CODE
2873  012402  012737  000001  000712          MOV     #1,EXFL
2874  012410  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2875  012414  000410                          BR      LT41X
2876  012416  012701  002100          4$:     MOV     #2100,R1        ;SET EXPT ERROR BITS
2877  012422  017702  166076                  MOV     @ER,R2          ;GET ERROR REGISTER
2878  012426  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2879  012430  001402                          BEQ     LT41X           ;IF NOT: BR
2880  012432  004737  015144                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2881  012436  005002                  LT41X:  CLR     R2              ;SET TIMER
2882  012440  032777  000200  166054  1$:     BIT     #200,@DS        ;SEE IF DRY SET
2883  012446  001002                          BNE     2$              ;IF SO: BR
2884  012450  005302                          DEC     R2              ;AWAIT DRY
2885  012452  001372                          BNE     1$              ;DELAY
2886  012454  004737  016534          2$:     JSR     PC,ITER         ;GO SEE IF ITERATIONS
2887  012460  004737  015624                  JSR     PC,DRVCLR       ;GO DO DRIVE CLEAR
2888  012464  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2889
2890                                        ;THE FOLLOWING SIX(6) TEST WILL REQUIRE TAPE MOVEMENT.  EACH
2891                                        ;TEST WILL PERFORM A TAPE WRITE WHILE IN A PARITICULAR MAINTENANCE
2892                                        ;MODE IN ORDER TO FORCE THE REMAINING ERROR CONDITIONS.
2893
2894                                        ;LOGIC TEST 42: CYCLIC REDUNDANCY ERROR(CRC)**********
2895
2896  012470  012737  001700  000772  LT42:   MOV     #1700,UDES      ;SET UNIT DESCRIPTION = NRZ
2897  012476  004737  014760                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
2898  012502  012700  001000                  MOV.    #1000,R0
2899  012506  005300                  1$:     DEC     R0
2900  012510  001376                          BNE     1$              ;PAUSE
2901  012512  012737  026242  000616          MOV     #MSLT42,EMADDR
2902  012520  012737  012526  000706          MOV     #LT42IT,SCOLP   ;SET SCOPE ADDRESS
2903  012526  004737  016740          LT42IT: JSR     PC,INIT         ;INIT SELECT DRIVE+SLAVE
2904  012532  012777  177770  165752          MOV     #-10,@WC
2905  012540  012777  177760  165750          MOV     #-20,@FC        ;SET FC=20
2906  012546  012777  027776  165740          MOV     #WDATA,@BA      ;SET BUS ADDRESS
2907  012554  012777  000007  165752          MOV     #7,@MR          ;SET MM CODE
2908  012562  012777  000061  165720          MOV     #61,@C1         ;LOAD WRITE+GO
2909  012570  005000                          CLR     R0
2910  012572  032777  000200  165722  LT42A:  BIT     #200,@DS        ;SEE IF DRY=1
2911  012600  001002                          BNE     LT42B           ;IF SO: BR
2912  012602  005300                          DEC     R0
2913  012604  001372                          BNE     LT42A           ;DELAY
2914  012606  022777  000200  165710  LT42B:  CMP     #200,@ER        ;SEE IF LRC ERROR ONLY
2915  012614  001007                          BNE     LT42B1          ;IF NOT: BR
2916  012616  017702  165706                  MOV     @CC,R2          ;GET CHECK CHAR
2917  012622  042702  177000                  BIC     #177000,R2      ;MASK CRC
2918  012626  022702  000777                  CMP     #777,R2         ;SEE IF SETUP CRC IS CORRECT
2919  012632  001410                          BEQ     LT42B2          ;IF SO: BR
2920  012634  004737  015144          LT42B1: JSR     PC,LTGER3       ;ELSE PRINT ERROR SETUP
2921  012640  012704  023034                  MOV     #MSG55,R4
2922  012644  004737  017662                  JSR     PC,TTOU         ;PRINT SETUP ERROR MSG
2923  012650  000137  002460                  JMP     TSCD2           ;RETURN TO SCHED
2924  012654  004737  016740          LT42B2: JSR     PC,INIT         ;GO INIT
2925  012660  012777  177770  165624          MOV     #-10,@WC        ;SET WC
2926  012666  012777  177760  165622          MOV     #-20,@FC        ;SET FC
2927  012674  012777  027776  165612          MOV     #WDATA,@BA      ;SET BA
2928  012702  012777  000021  165624          MOV     #21,@MR         ;SET MM
2929  012710  012777  000061  165572          MOV     #61,@C1         ;LOAD WRITE+GO
2930  012716  005000                          CLR     R0
2931  012720  032777  000200  165574  LT42C:  BIT     #200,@DS        ;SEE IF DRY
2932  012726  001002                          BNE     LT42D           ;IF SO: BR
2933  012730  005300                          DEC     R0
2934  012732  001372                          BNE     LT42C           ;AWAIT DRY
2935  012734  005777  165564          LT42D:  TST     @ER             ;SEE IF CRC=1
2936  012740  100411                          BMI     LT42E           ;IF SO: BR
2937  012742  012737  027745  000666          MOV     #TMS36,ERADD    ;SET ERROR CODE
2938  012750  012737  000001  000712          MOV     #1,EXFL
2939  012756  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
2940  012762  000410                          BR      LT42X
2941  012764  012701  100200          LT42E:  MOV     #100200,R1      ;SET EXPT ERROR BITS
2942  012770  017702  165530                  MOV     @ER,R2          ;GET ERROR REGISTER
2943  012774  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2944  012776  001402                          BEQ     LT42X           ;IF NOT: BR
```

```
2945  013000  004737  015144                LT42X:  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2946  013004  004737  016534                        JSR     PC,ITER         ;DO ITERATIONS
2947  013010  004737  015624                        JSR     PC,DRVCLR
2948  013014  000137  002460                        JMP     TSCD2           ;RETURN TO SCHED
2949
2950                                              ;LOGIC TEST 43: LONGITUALINAL REDUNDANCY(LRC)**********
2951
2952  013020  032777  000004  165510  LT43:   BIT     #4,@DT          ;++B BRANCH IF NOT A TE16
2953  013026  001114                          BNE     LT43XX          ;++B
2954  013030  012737  001700  000772          MOV     #1700,UDES      ;SET UNIT DESCRIPTION = NRZ
2955  013036  004737  014760                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
2956  013042  012737  013056  000706          MOV     #LT43IT,SCOLP   ;SET SCOPE ADDRESS
2957  013050  012737  026276  000616          MOV     #MSLT43,EMADDR
2958  013056  004737  016740          LT43IT: JSR     PC,INIT         ;INIT, SELECT DRIVE+SLAVE
2959  013062  005001                          CLR     R1
2960  013064  005301                  1$:     DEC     R1              ;DELAY
2961  013066  001376                          BNE     1$
2962  013070  012777  000023  165436          MOV     #23,@MR         ;SET MM
2963  013076  012777  177770  165406          MOV     #-10,@WC        ;SET WC
2964  013104  012777  177760  165404          MOV     #-20,@FC        ;SET FC
2965  013112  012777  027776  165374          MOV     #WDATA,@BA      ;SET BA
2966  013120  012777  000061  165362          MOV     #61,@C1         ;LOAD WRITE+GO
2967  013126  005000                          CLR     R0
2968  013130  032777  000200  165364  LT43C:  BIT     #200,@DS        ;SEE IF DRY
2969  013136  001002                          BNE     LT43D           ;IF SO: BR
2970  013140  005300                          DEC     R0
2971  013142  001372                          BNE     LT43C           ;AWAIT DRY
2972  013144  032777  000200  165352  LT43D:  BIT     #200,@ER        ;SEE IF LRC=1
2973  013152  001011                          BNE     LT43E           ;IF SO: BR
2974  013154  012737  027535  000666          MOV     #TMS26,ERADD    ;SET ERROR CODE
2975  013162  012737  000001  000712          MOV     #1,EXFL
2976  013170  004737  015156                  JSR     PC,LTGER0       ;GO PRINT
2977  013174  000425                          BR      LT43X
2978  013176  017702  165332          LT43E:  MOV     @MR,R2
2979  013202  042702  000177                  BIC     #177,R2         ;MASK LRC
2980  013206  012701  157600                  MOV     #157600,R1      ;SET EXPT LRC
2981  013212  020102                          CMP     R1,R2           ;SEE IF EXPT = RCVD
2982  013214  001405                          BEQ     LT43F           ;IF SO: BR
2983  013216  012737  023011  000666          MOV     #MSG53,ERADD    ;SET ERROR CODE
2984  013224  004737  016272                  JSR     PC,LTGER1       ;PRINT ERROR
2985  013230  017702  165270          LT43F:  MOV     @ER,R2          ;GET ERROR REGISTER
2986  013234  012701  000200                  MOV     #200,R1         ;SET EXPT ERROR BITS
2987  013240  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2988  013242  001402                          BEQ     LT43X           ;IF NOT: BR
2989  013244  004737  015144                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2990  013250  004737  016534          LT43X:  JSR     PC,ITER
2991  013254  004737  015624                  JSR     PC,DRVCLR
2992  013260  000137  002460          LT43XX: JMP     TSCD2           ;RETURN TO SCHED
```

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

013264  012737  002300  000772  LT44:   MOV     #2300,UDES      ;SET UNIT DESRIPTION = PE
013272  004737  014760          LT44:   JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
013276  012737  026332  000616          MOV     #MSLT44,EMADDR  ;SET HEADER
013304  012737  013312  000706          MOV     #LT44IT,SCOLP   ;SET SCOP
013312  004737  016740          LT44IT: JSR     PC,INIT         ;GO INITIALIZE
013316  012777  177600  165166          MOV     #-200,@WC       ;SET WC=200
013324  012777  177400  165164          MOV     #-400,@FC       ;SET FC=400
013332  012777  027776  165154          MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
013340  012777  000061  165142          MOV     #61,@C1         ;LOAD WRITE AND GO
013346  005000                          CLR     R0
013350  005777  165142          LT44A:  TST     @FC             ;SEE IF FC=0
013354  001402                          BEQ     LT44A1          ;IF SO:BR
013356  005300                          DEC     R0
013360  001373                          BNE     LT44A           ;AWAIT FC=0
013362  012777  000021  165144  LT44A1: MOV     #21,@MR         ;SET MAINT MODE
013370  005000                          CLR     R0
013372  032777  000200  165122  LT44B:  BIT     #200,@DS        ;SEE IF DRY
013400  001002                          BNE     LT44C           ;IF SO :BR
013402  005300                          DEC     R0
013404  001372                          BNE     LT44B           ;AWAIT DRY
013406  005777  165112          LT44C:  TST     @ER             ;SEE IF CORR=1
013412  100410                          BMI     LT44D           ;IF SO: BR
013414  012737  027736  000666          MOV     #TMS35,ERADD    ;ELSE SET ERROR CODE
013422  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLAG
013430  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
013434  000240          LT44D:  NOP
013436  122777  000002  165064  LT44E:  CMPB    #2,@CC          ;SEE IF DEAD TRACK BIT 1
013444  001414                          BEQ     LT44F           ;IF SO: BR
013446  117702  165056                  MOVB    @CC,R2          ;ELSE SAVE RECVD
013452  042702  177000                  BIC     #177000,R2      ;MASK OUT CRC
013456  112701  000002                  MOVB    #2,R1           ;SAVE EXPT
013462  012737  022420  000666          MOV     #MSG42,ERADD    ;SET ERROR CODE
013470  004737  016272                  JSR     PC,LTGER1       ;GO PRINT ERROR
013474  000410                          BR      LT44X
013476  017702  165022          LT44F:  MOV     @ER,R2          ;GET ERROR REGISTER
013502  012701  100000                  MOV     #100000,R1      ;SET EXPT ERROR BITS
013506  020102                          CMP     R1,R2           ;SEE IF EXPT=RCVD
013510  001402                          BEQ     LT44X           ;IF SO: BR
013512  004737  015144                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
013516  004737  016534          LT44X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
013522  004737  015624                  JSR     PC,DRVCLR       ;GO DO DRIVE CLEAR
013526  000137  002460          LT44XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
3037
3038                                        ;LOGIC TEST 45: PE INCORRECTABLE DATA(INC)**********
3039
3040  013532  012737  002300  000772  LT45:   MOV     #2300,UDES      ;SET UNIT DESCRIPTION = PE
3041  013540  004737  014760                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
3042  013544  012737  026412  000616          MOV     #MSLT45,EMADDR
3043  013552  012737  013560  000706          MOV     #LT45IT,SCOLP
3044  013560  004737  016740          LT45IT: JSR     PC,INIT         ;INIT SELECT DRIVE SLAVE
3045  013564  012777  177600  164720          MOV     #-200,@WC       ;SET WC=200
3046  013572  012777  177400  164716          MOV     #-400,@FC       ;SET FC=400
3047  013600  012777  027776  164706          MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
3048  013606  012777  000061  164674          MOV     #61,@CS         ;LOAD WRITE+GO
3049  013614  005000                          CLR     R0
3050  013616  005777  164674          LT45E:  TST     @FC             ;AWAIT FC=0
3051  013622  001402                          BEQ     LT45E1
3052  013624  005300                          DEC     R0
3053  013626  001373                          BNE     LT45E           ;AWAIT FC 0
3054  013630  012777  000023  164676  LT45E1: MOV     #23,@MR         ;SET MAINT CODE
3055  013636  005000                          CLR     R0
3056  013640  032777  000200  164654  LT45A:  BIT     #200,@DS        ;SEE IF DRY IS SET
3057  013646  001002                          BNE     LT45B           ;IF SO: BR
3058  013650  005300                          DEC     R0
3059  013652  001372                          BNE     LT45A           ;AWAIT DRY
3060  013654  032777  000100  164642  LT45B:  BIT     #100,@ER        ;SEE IF INC-1
3061  013662  001010                          BNE     LT45D           ;IF SO:BR
3062  013664  012737  027513  000666          MOV     #TMS23,ERADD    ;SET ERROR CODE
3063  013672  012737  000001  000712          MOV     #1,EXFL
3064  013700  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
3065  013704  017702  164620          LT45D:  MOV     @CC,R2          ;GET CHECK CHAR
3066  013710  042702  177000                  BIC     #177000,R2      ;MASK CHECK CHAR
3067  013714  012701  000046                  MOV     #46,R1          ;SET EXPT CK
3068  013720  020102                          CMP     R1,R2           ;SEE IF EXPT = RCVD
3069  013722  001405                          BEQ     LT45F           ;IF SO: BR
3070  013724  012737  023023  000666          MOV     #MSG54,ERADD
3071  013732  004737  016272                  JSR     PC,LTGER1       ;ELSE GO PRINT ERROR
3072  013736  017702  164562          LT45F:  MOV     @ER,R2
3073  013742  042702  120600                  BIC     #120600,R2      ;MASK OPI ,NSG, CORR, AND PEF
3074  013746  012701  000100                  MOV     #100,R1         ;SET EXPT ERROR BITS
3075  013752  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
3076  013754  001402                          BEQ     LT45X           ;IF NOT: BR
3077  013756  004737  015144                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
3078  013762  004737  016534          LT45X:  JSR     PC,ITER
3079  013766  004737  015624                  JSR     PC,DRVCLR
3080  013772  000137  002460          LT45XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
3081
3082                                        ;LOGIC TEST 46: PE FORMAT ERROR(PEF,NSG)**********
3083
3084  013776  012737  002300  000772  LT46:   MOV     #2300,UDES      ;SET UNIT DESCRIPTION = PE
3085  014004  004737  014760                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
3086  014010  012737  026474  000616          MOV     #MSLT46,EMADDR  ;SET HEADER
3087  014016  012737  014024  000706          MOV     #LT46IT,SCOLP   ;SET SCOPE ADDRESS
3088  014024  004737  016740          LT46IT: JSR     PC,INIT         ;INITIALIZE
3089  014030  012777  177770  164454          MOV     #-10,@WC        ;SET WC=10
3090  014036  012777  177760  164452          MOV     #-20,@FC        ;SET FC=20
3091  014044  012777  027776  164442          MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
3092  014052  012777  000061  164430          MOV     #61,@CS         ;LOAD WRITE+GO
3093  014060  005777  164432          LT46A:  TST     @FC
3094  014064  001375                          BNE     LT46A           ;AWAIT FC=0
3095  014066  032777  000100  164440  1$:     BIT     #100,@MR        ;WAIT FOR TAPE TO START WRITEING POSTAMBLE
3096  014074  001774                          BEQ     1$              ;DELAY
3097  014076  032777  000100  164430  2$:     BIT     #100,@MR
3098  014104  001374                          BNE     2$
3099  014106  012777  000027  164420          MOV     #27,@MR         ;SET MM CODE TO KILL PEF
3100  014114  005000                          CLR     R0              ;INIT TIMING LOOP
3101  014116  032777  000200  164376  LT46B:  BIT     #200,@DS        ;SEE IF DRY SET
3102  014124  001002                          BNE     LT46C           ;IF SO: BR
3103  014126  005300                          DEC     R0
3104  014130  001372                          BNE     LT46B           ;AWAIT DRY
3105  014132  032777  000200  164364  LT46C:  BIT     #200,@ER        ;SEE IF PEF SET
3106  014140  001011                          BNE     LT46D           ;IF SO: BR
3107  014142  012737  027527  000666          MOV     #TMS25,ERADD    ;SET ERROR TAG
3108  014150  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLAG
3109  014156  004737  015156                  JSR     PC,LTGER0       ;GO PRINT ERROR
3110  014162  000420                          BR      LT46X
3111  014164  017702  164334          LT46D:  MOV     @ER,R2          ;GET ERROR REGISTER
3112  014170  042702  120100                  BIC     #120100,R2      ;++B CLEAR CRC,OPI & INC BITS (MAY OR MAY NOT SET)
3113  014174  012701  000600                  MOV     #600,R1         ;++B SET EXPT ERROR BITS (NSG + PEF)
3114  014200  032777  000004  164330          BIT     #4,@DT          ;++B BRANCH IF TE16
3115  014206  001402                          BEQ     1$              ;++B
3116  014210  042701  000400                  BIC     #400,R1         ;++B TU77 SHOULD NOT SET NSG BIT
3117  014214  020102                  1$:     CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
3118  014216  001402                          BEQ     LT46X           ;IF NOT: BR
3119  014220  004737  015144                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
3120  014224  004737  016534          LT46X:  JSR     PC,ITER
3121  014230  004737  015624                  JSR     PC,DRVCLR
3122  014234  000137  002460          LT46XX: JMP     TSTD2           ;RETURN TO SCHED
```

```
3123                                            ;LOGIC TEST 47: FRAME COUNT OVERFLOW(M8905-YB)*******
3124
3125  014240  012737  026530  000616  LT47:   MOV     #MSLT47,EMADDR  ;SET TEST HEADER
3126  014246  012737  014254  000706          MOV     #LT47IT,SCOLP   ;SET SCOPE ADDRESS
3127  014254  004737  016710          LT47IT: JSR     PC,INIT3        ;GO INIT
3128  014260  012777  177770  164224          MOV     #-10,@WC        ;SET WC = 10
3129  014266  012777  177760  164222          MOV     #-20,@FC        ;SET FC = 20
3130  014274  052777  001700  164240          BIS     #1700,@TC       ;SET TO NRZ, NORMAL, ODD
3131  014302  012777  027776  164204          MOV     #WDATA,@BA      ;SET BUS ADDRESS
3132  014310  012777  000013  164216          MOV     #13,@MR         ;SET WRAP 2
3133  014316  012777  000061  164164          MOV     #61,@C1         ;LOAD WRITE+GO
3134  014324  012700  040000                  MOV     #40000,R0
3135  014330  005777  164206          LT47A:  TST     @TC             ;SEE IF ALPHA
3136  014334  100002                          BPL     LT47B           ;IF SO: BR
3137  014336  005300                          DEC     R0
3138  014340  001373                          BNE     LT47A           ;AWAIT ALPHA
3139  014342  012700  000020          LT47B:  MOV     #20,R0          ;SET CLK CNT
3140  014346  052777  000040  164160  LT47C:  BIS     #40,@MR
3141  014354  042777  000040  164152          BIC     #40,@MR         ;CLOCK MR
3142  014362  005300                          DEC     R0
3143  014364  001370                          BNE     LT47C           ;IF NOT DONE ALL: BR
3144  014366  017702  164124                  MOV     @FC,R2
3145  014372  005001                          CLR     R1              ;SET TEST WORD
3146  014374  020102                          CMP     R1,R2           ;SEE IF EXPT = RCVD
3147  014376  001410                          BEQ     LT47X           ;IF SO: BR
3148  014400  012737  021673  000666          MOV     #MSG19,ERADD    ;SET ERROR CODE
3149  014406  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLAG
3150  014414  004737  016272                  JSR     PC,LTGER1       ;GO PRINT ERROR
3151  014420  004737  016534          LT47X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
3152  014424  000137  002460                  JMP     @#SCD2          ;RETURN TO SCHEDULAR
3153
```

```
3154
3155                                                ;LOGIC TEST 50: NEF WHEN WRITING PE ON NRZ SELECTED SLAVE
3156
3157  014430  012737  026600  000616  LT50:   MOV     #MSLT50,EMADDR  ;SET ERROR POINTER FOR STANDARD
3158  014436  012737  014444  000706          MOV     #LT50IT,SCOLP
3159  014444  004737  016710          LT50IT: JSR     PC,INIT3        ;SET SLAVE = NRZ
3160  014450  042777  003400  164064          BIC     #3400,@TC       ;CLEAR DENSITY BITS
3161  014456  052777  002300  164056          BIS     #2300,@TC       ;SET DENSITY = PE
3162  014464  012777  177770  164020          MOV     #-10,@WC        ;SET WORD COUNT
3163  014472  012777  177760  164016          MOV     #-20,@FC        ;SET FRAME COUNT
3164  014500  012777  027776  164006          MOV     #WDATA,@BA      ;SET BUS ADDRESS
3165  014506  012777  000013  164020          MOV     #13,@MR         ;SET WRAP 2
3166  014514  012777  000061  163766          MOV     #61,@C1         ;LOAD WRITE COMMAND
3167  014522  000240                          NOP
3168  014524  000240                          NOP
3169  014526  000240                          NOP
3170  014530  012701  004000          2$:     MOV     #4000,R1        ;SET EXPECTED RESULT
3171  014534  017702  163764          3$:     MOV     @ER,R2          ;GET ERROR REGISTER
3172  014540  030102                          BIT     R1,R2           ;BRANCH IF NEF BIT SET
3173  014542  001006                          BNE     1$
3174  014544  012737  000001  000712          MOV     #1,EXFL         ;SET EXPECTED FLAG
3175  014552  004737  015156                  JSR     PC,LTGER0       ;PRINT ERROR
3176  014556  000404                          BR      LT50X
3177  014560  020102                  1$:     CMP     R1,R2           ;BRANCH IF NO UNEXPECTED ERROR
3178  014562  001402                          BEQ     LT50X           ;BITS WERE SET
3179  014564  004737  015144                  JSR     PC,LTGER3       ;PRINT ERROR MSG
3180  014570  004737  016534          LT50X:  JSR     PC,ITER         ;+ITERATE TEST
3181  014574  004737  015624                  JSR     PC,DRVCLR       ;RESET DRIVE
3182  014600  000137  002460                  JMP     TSCD2
3183
3184
```

```
3185
3186                                        ;LOGIC TEST 51: NEF WHEN WRITING NRZ ON PE SELECTED SLAVE
3187
3188  014604  012737  026660  000616  LT51:   MOV     #MSLT51,EMADDR  ;+SET ERROR POINTER FOR STANDARD CONF.
3189  014612  012737  014620  000706          MOV     #LT51IT,SCOLP   ;SET SCOPE LOOP ADDRS.
3190  014620  004737  016724          LT51IT: JSR     PC,INIT4        ;SET SLAVE = PE
3191  014624  042777  002300  163710          BIC     #2300,@TC       ;CLEAR DENSITY BITS
3192  014632  052777  001300  163702          BIS     #1300,@TC       ;SET DENSITY = NRZ
3193  014640  012777  177770  163644          MOV     #-10,@WC        ;SET WORD COUNT
3194  014646  012777  177760  163642          MOV     #-20,@FC        ;SET FRAME COUNT
3195  014654  012777  027776  163632          MOV     #WDATA,@BA      ;SET BUS ADDRESS
3196  014662  012777  000013  163644          MOV     #13,@MR         ;SET WRAP 2
3197  014670  012777  000061  163612          MOV     #61,@C1         ;SET WRITE COMMAND AND GO
3198  014676  000240                          NOP
3199  014700  000240                          NOP
3200  014702  000240                          NOP
3201  014704  012701  004000                  MOV     #4000,R1        ;SET EXPECTED RESULT
3202  014710  017702  163610                  MOV     @ER,R2          ;GET ERROR REGISTER
3203  014714  030102                          BIT     R1,R2           ;BRANCH IF NEF SET
3204  014716  001006                          BNE     1$
3205  014720  012737  000001  000712          MOV     #1,EXFL         ;SET EXPECTED FLAG
3206  014726  004737  015156                  JSR     PC,LTGER0       ;PRINT ERROR MSG
3207  014732  000404                          BR      LT51X
3208  014734  020102                  1$:     CMP     R1,R2           ;BRANCH IF NO UNEXPECTED
3209  014736  001402                          BEQ     LT51X           ;ERROR BITS WERE SET
3210  014740  004737  015144                  JSR     PC,LTGER3
3211  014744  004737  016534          LT51X:  JSR     PC,ITER         ;+ITERATE TEST
3212  014750  004737  015624                  JSR     PC,DRVCLR       ;CLEAR DRIVE
3213  014754  000137  002460                  JMP     TSCD2           ;RETURN TO SCHEDULER
```

```
3214                                    ;STATIC TESTS ONLY SUBROUTINE*********
3215
3216  014760  005737  000734    STATIC: TST     STFLG           ;SEE IF SINGLE TEST ONLY
3217  014764  001006                    BNE     1$              ;IF SO: BR
3218  014766  005737  001006            TST     STATC           ;SEE IF STATIC ONLY
3219  014772  001403                    BEQ     1$              ;IF NOT: BR
3220  014774  005726                    TST     (SP)+           ;RESET STACK
3221  014776  000137  002460            JMP     TSCD2           ;RETURN TO SCHEDULAR
3222  015002  005037  001002    1$:     CLR     RDRVF
3223  015006  000207                    RTS     PC              ;RETURN TO TEST
3224
```

```
3225
3226                                        ;END OF RECORD FORCE SUBROUTINE*****
3227
3228  015010  017700  163520        EORPA:  MOV     @MR,R0          ;GET MAINT REG
3229  015014  042700  000036                BIC     #36,R0          ;CLEAR CURRENT OP CODE
3230  015020  052700  000024                BIS     #24,R0          ;SET EOR CLEAR OP CODE
3231  015024  010077  163504                MOV     R0,@MR          ;DO EOR
3232  015030  042777  000037  163476        BIC     #37,@MR         ;CLEAR EOR AND MM
3233  015036  005000                        CLR     R0
3234  015040  012701  000002                MOV     #2,R1
3235  015044  032777  000001  163436 EORP1:  BIT     #1,@C1          ;SEE IF GO GONE
3236  015052  001430                        BEQ     EORP2           ;IF SO: BR
3237  015054  005300                        DEC     R0
3238  015056  001372                        BNE     EORP1           ;AWAIT GO RESET
3239  015060  005301                        DEC     R1
3240  015062  001370                        BNE     EORP1
3241  015064  032777  020000  163476        BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
3242  015072  001020                        BNE     EORP2           ;IF SO: BR
3243  015074  005737  000614               TST     HDRFL           ;SEE IF DONE HEADER
3244  015100  001004                        BNE     EORP1A          ;IF SO: BR
3245  015102  013704  000616                MOV     EMADDR,R4
3246  015106  004737  017662                JSR     PC,TTOUT        ;PRINT HEADER
3247  015112  012704  030411        EORP1A: MOV     #WMSG31,R4
3248  015116  004737  017662                JSR     PC,TTOUT        ;PRINT EOR GO BIT ERROR
3249  015122  032777  100000  163440        BIT     #100000,@SWR    ;SEE IF HALT ON ERROR
3250  015130  001401                        BEQ     EORP2           ;IF NOT: BR
3251  015132  000000                        HALT
3252  015134  000240        EORP2:  NOP
3253  015136  005037  000672        ECRPX:  CLR     TEMP2           ;CLEAR FLAG
3254  015142  000207                        RTS     PC              ;RETURN
3255
```

```
3256                                        ;LOGIC TEST ADDRESSING ERROR SUBROUTINE**********
3257
3258  015144  005037  000712          LTGER3: CLR     EXFL
3259  015150  012737  022762  000666          MOV     #MSG51,ERADD
3260  015156  012737  000001  000742  LTGER0: MOV     #1,ADDFL        ;SET NO ADDRESS FLAG
3261  015164  000240                  LTGER:  NOP
3262  015166  005037  000662                  CLR     PFLG            ;CLEAR PRINT FLAG
3263  015172  032777  020000  163370          BIT     #20000,@SWR     ;SEE IF SHOULD PRINT
3264  015200  001112                          BNE     LTGX            ;IF NOT: BR
3265  015202  005737  000614          LTGA:   TST     HDRFL           ;SEE IF PRINTED HEADER
3266  015206  001004                          BNE     LTGA1           ;IF SO: BR
3267  015210  013704  000616                  MOV     EMADDR,R4
3268  015214  004737  017662                  JSR     PC,TTOUT        ;PRINT TEST HEADER
3269  015220  012737  000001  000614  LTGA1: MOV     #1,HDRFL        ;SET HEADER FLAG
3270  015226  013704  000666                  MOV     ERADD,R4
3271  015232  004737  017662                  JSR     PC,TTOUT        ;PRINT CONDITION ERROR
3272  015236  005737  000742                  TST     ADDFL
3273  015242  001003                          BNE     LTGA2
3274  015244  010103                          MOV     R1,R3
3275  015246  004737  020010                  JSR     PC,OCTP         ;PRINT ADDRESS
3276  015252  005737  000712          LTGA2:  TST     EXFL
3277  015256  001412                          BEQ     LTGC            ;IF NO STATUS: BR
3278  015260  012704  021253                  MOV     #MSG6,R4
3279  015264  022737  000001  000712          CMP     #1,EXFL         ;EXPT-NOT RCVD
3280  015272  001402                          BEQ     LTGB
3281  015274  012704  021272                  MOV     #MSG7,R4        ;RCVD-NOT EXPT
3282  015300  004737  017662          LTGB:   JSR     PC,TTOUT        ;PRINT STATUS
3283  015304  005237  000662          LTGC:   INC     PFLG
3284  015310  005737  000742                  TST     ADDFL           ;SEE IF ADD TST
3285  015314  001430                          BEQ     LTGD            ;IF SO: BR
3286  015316  005737  000740                  TST     T24FL           ;SEE IF TEST 24
3287  015322  001423                          BEQ     LTGC0           ;IF NOT: BR
3288  015324  012704  030376                  MOV     #WMSG27,R4
3289  015330  004737  017662                  JSR     PC,TTOUT        ;PRINT DATA TAG
3290  015334  012704  021573                  MOV     #MSG12,R4
3291  015340  004737  017662                  JSR     PC,TTOUT        ;PRINT EXPT TAG
3292  015344  012703  177777                  MOV     #-1,R3
3293  015350  004737  020000                  JSR     PC,OCTPE        ;PRINT EXPT
3294  015354  012704  021602                  MOV     #MSG13,R4
3295  015360  004737  017662                  JSR     PC,TTOUT        ;PRINT RCVD TAG
3296  015364  010103                          MOV     R1,R3           ;GET RCVD
3297  015366  004737  020000                  JSR     PC,OCTPE        ;PRINT RCVD
3298  015372  004737  015470          LTGC0:  JSR     PC,REGP         ;PRINT REGISTERS
3299  015376  032777  004000  163164  LTGD:   BIT     #4000,@SWR
3300  015404  001010                          BNE     LTGX
3301  015406  012704  021644                  MOV     #MSG16,R4
3302  015412  004737  017662                  JSR     PC,TTOUT
3303  015416  013703  000676                  MOV     ITCNT,R3        ;PRINT ITERATION
3304  015422  004737  020010                  JSR     PC,OCTP
3305  015426  005777  163136          LTGX:   TST     @SWR
3306  015432  100001                          BPL     LTGXA           ;IF NOT STOP ON ERROR: BR
3307  015434  000000                          HALT
3308  015436  005737  000662          LTGXA:  TST     PFLG
3309  015442  001004                          BNE     LTGXX           ;IF PRINTED: BR
3310  015444  032777  020000  163116          BIT     #20000,@SWR
3311  015452  001653                          BEQ     LTGA
```

```
3312  015454  005037  000742          LTGXX:  CLR     ADDFL           ;CLEAR ADDRESS FLAG
3313  015460  005037  000712                  CLR     EXFL
3314  015464  000137  016504                  JMP     SCOPE
3315
3316                                          ;SUBROUTINE TO PRINT MAJOR REGISTERS**********
3317
3318  015470  000240                  REGP:   NOP
3319  015472  012704  022605                  MOV     #MSG46,R4
3320  015476  004737  017662                  JSR     PC,TTOUT        ;PRINT REGISTER HEADER
3321  015502  017703  163002                  MOV     @C1,R3
3322  015506  004737  020000                  JSR     PC,OCTPE
3323  015512  017703  162774                  MOV     @WC,R3
3324  015516  004737  020000                  JSR     PC,OCTPE
3325  015522  017703  162766                  MOV     @BA,R3
3326  015526  004737  020000                  JSR     PC,OCTPE
3327  015532  017703  162760                  MOV     @FC,R3
3328  015536  004737  020000                  JSR     PC,OCTPE
3329  015542  017703  162752                  MOV     @CS,R3
3330  015546  004737  020000                  JSR     PC,OCTPE
3331  015552  017703  162744                  MOV     @DS,R3
3332  015556  004737  020000                  JSR     PC,OCTPE                ;PRINT REGISTERS
3333  015562  017703  162736                  MOV     @ER,R3
3334  015566  004737  020000                  JSR     PC,OCTPE
3335  015572  017703  162730                  MOV     @AS,R3
3336  015576  004737  020000                  JSR     PC,OCTPE
3337  015602  017703  162726                  MOV     @MR,R3
3338  015606  004737  020000                  JSR     PC,OCTPE
3339  015612  017703  162724                  MOV     @TC,R3
3340  015616  004737  020000                  JSR     PC,OCTPE
3341  015622  000207                  RTS     PC
3342
3343
```

```
3344                                                ;DRIVE CLEAP SUBROUTINE*********
3345
3346  015624  000240                        DRVCLR: NOP
3347  015626  012704  040000                        MOV     #40000,R4
3348  015632  005304                        DCD:    DEC     R4
3349  015634  001376                                BNE     DCD             ;DELAY
3350  015636  005037  000662                        CLR     PFLG
3351  015642  004737  016060                        JSR     PC,ATTN         ;GO SEE OF ATTN SET
3352  015646  012777  000011  162634              MOV     #11,@C1         ;ISSUE DRIVE CLEAR
3353  015654  005000                                CLR     R0
3354  015656  032777  000200  162636  DCA:    BIT     #200,@DS        ;SEE IF DRY
3355  015664  001002                                BNE     DCA0
3356  015666  005300                                DEC     R0
3357  015670  001372                                BNE     DCA             ;WAIT FOR DRY
3358  015672  032777  040000  162622  DCA0:   BIT     #40000,@DS      ;SEE IF ERR RESET
3359  015700  001022                                BNE     DCE             ;IF NOT: BR
3360  015702  005777  162616                        TST     @ER             ;SEE IF ERROR REGISTER RESET
3361  015706  001017                                BNE     DCE             ;IF NOT: BR
3362  015710  005777  162606                        TST     @DS             ;SEE IF ATA RESET
3363  015714  100414                                BMI     DCE             ;IF NOT: BR
3364  015716  012703  000001                        MOV     #1,R3           ;SET TEST BIT
3365  015722  013704  000620                        MOV     DRVN,R4         ;GET DRIVE NUMBER & BRANCH
3366  015726  001403                                BEQ     DCC             ;IF DRIVE 0
3367  015730  006303                        DCB:    ASL     R3              ;POSITION TEST BIT PER DRIVE NUMBER
3368  015732  005304                                DEC     R4              ;SEE IF DONE
3369  015734  001375                                BNE     DCB             ;IF NOT: BR
3370  015736  030377  162564                DCC:    BIT     R3,@AS          ;SEE IF ATTEN IS RESET
3371  015742  001001                                BNE     DCE             ;IF NOT: BR
3372  015744  000207                                RTS     PC              ;RETURN
3373
3374  015746  000240                        DCE:    NOP
3375  015750  032777  020000  162612              BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
3376  015756  001017                                BNE     DCEX            ;IF SO: BR
3377  015760  005737  000614                        TST     HDRFL           ;SEE IF PRINT HEADER
3378  015764  001004                                BNE     DCEA            ;IF NOT: BR
3379  015766  013704  000616                        MOV     EMADDR,R4
3380  015772  004737  017662                        JSR     PC,TTOUT        ;PRINT HEADER
3381  015776  012704  022711                DCEA:   MOV     #MSG47,R4
3382  016002  004737  017662                        JSR     PC,TTOUT        ;PRINT DRIVE CLEAR ERROR
3383  016006  004737  015470                        JSR     PC,REGP         ;PRINT REGISTERS
3384  016012  005237  000662                        INC     PFLG            ;SET PRINTED FLAG
3385  016016  005777  162546                DCEX:   TST     @SWR            ;SEE IF HALT ON ERROR
3386  016022  100001                                BPL     DCEXA           ;IF NOT: BR
3387  016024  000000                                HALT
3388  016026  005737  000662                DCEXA:  TST     PFLG            ,SEE IF HAVE PRINTED
3389  016032  001004                                BNE     DCEXX           ;IF SO: BR
3390  016034  032777  020000  162526                BIT     #20000,@SWR     ;BRANCH IF ERROR
3391  016042  001741                                BEQ     DCE             ;PRINTOUT DESIRED
3392  016044  000240                        DCEXX:  NOP
3393  016046  012737  015624  000706                MOV     #DRVCLR,SCOPE   ;SET SCOPE LOOP ADDRESS
3394  016054  000137  016504                        JMP     SCOPE           ;GO DO SCOPE LOOP
```

```
3395                                                ;COMPOSITE ERROR CHECK SUBROUTINE**********
3396
3397  016060  000240                        ATTN:   NOP
3398  016062  005777  162434                        TST     @DS             ;SEE IF ATA SET
3399  016066  001004                                BNE     ATTA            ;IF SO: BR
3400  016070  012737  022247  000674                MOV     #MSG32,TEMP3
3401  016076  000427                                BR      ATTP            ;ELSE PRINT ERROR
3402  016100  032777  040000  162414        ATTA:   BIT     #40000,@DS      ;SEE IF COMPOSITE ERROR SET
3403  016106  001004                                BNE     ATTB            ;IF SO: BR
3404  016110  012737  022231  000674                MOV     #MSG31,TEMP3
3405  016116  000417                                BR      ATTP            ;ELSE PRINT ERROR
3406  016120  012703  000001                ATTB:   MOV     #1,R3           ;SET TEST BIT
3407  016124  012737  022265  000674                MOV     #MSG33,TEMP3
3408  016132  013704  000620                        MOV     DRVN,R4         ;GET DRIVE NUMBER & BRANCH
3409  016136  001403                                BEQ     ATTD            ;IF DRIVE 0
3410  016140  006303                        ATTC:   ASL     R3              ;POSITION TEST BIT
3411  016142  005304                                DEC     R4              ;SEE IF DONE
3412  016144  001375                                BNE     ATTC            ;IF NOT: BR
3413  016146  030377  162354                ATTD:   BIT     R3,@AS          ;SEE IF ATTEN SUMMARY SET
3414  016152  001401                                BEQ     ATTP            ;IF NOT: BR
3415  016154  000207                                RTS     PC              ;ELSE RETURN
3416  016156  032777  020000  162404        ATTP:   BIT     #20000,@SWR     ;SEE IF PRINT INHIBIT
3417  016164  001021                                BNE     ATTX            ;IF SO: BR
3418  016166  005737  000614                        TST     HDRFL           ;SEE IF DONE HEADER
3419  016172  001004                                BNE     ATTPA           ;IF SO: BR
3420  016174  013704  000616                        MOV     EMADDR,R4
3421  016200  004737  017662                        JSR     PC,TTOUT        ;PRINT HEADER
3422  016204  013704  000674                ATTPA:  MOV     TEMP3,R4
3423  016210  004737  017662                        JSR     PC,TTOUT        ;PRINT ERROR TYPE
3424  016214  004737  015470                        JSR     PC,REGP         ;PRINT REGISTERS
3425  016220  005237  000662                        INC     PFLG            ;SET PRINT FLAG
3426  016224  005237  000614                        INC     HDRFL           ;SET HEADER FLAG
3427  016230  005777  162334                ATTX:   TST     @SWR            ;SEE IF HALT ON ERROR
3428  016234  100001                                BPL     ATTXA           ;IF NOT: BR
3429  016236  000000                                HALT
3430  016240  005737  000662                ATTXA:  TST     PFLG            ;SEE IF DONE PRINT
3431  016244  001004                                BNE     ATTXX           ;IF SO: BR
3432  016246  032777  020000  162374                BIT     #20000,@SWR     ;BRANCH IF NO ERROR
3433  016254  001740                                BEQ     ATTP            ;PRINTOUT DESIRED
3434  016256  005037  000662                ATTXX:  CLR     PFLG            ;CLEAR PRINT FLAG
3435  016262  000207                                RTS     PC              ;RETURN
```

```
3436                                        ;LOGIC TEST REGISTER BIT ERROR SUBROUTIINE**********
3437
3438  016264  012737  000001  000732  LTGER2: MOV     #1,PEXFL        ;SET FLAG
3439  016272  000240                  LTGER1: NOP
3440  016274  005037  000662                  CLR     PFLG            ;CLEAR PRINT FLAG
3441  016300  032777  020000  162262          BIT     #20000,@SWR     ;BRANCH IF ERROR
3442  016306  001055                          BNE     LTG1X           ;PRINTOUT DESIRED
3443  016310  005737  000614          LTG1A:  TST     HDRFL           ;SEE IF PRINT HEADER
3444  016314  001004                          BNE     LTG1B           ;IF NOT: BR
3445  016316  013704  000616                  MOV     EMADDR,R4
3446  016322  004737  017662                  JSR     PC,TTOUT        ;PRINT HEADER
3447  016326  012737  000001  000614  LTG1B:  MOV     #1,HDRFL        ;SET FLAG
3448  016334  013704  000666                  MOV     ERADD,R4
3449  016340  004737  017662                  JSR     PC,TTOUT        ;PRINT ERROR CODE
3450  016344  005737  000732                  TST     PEXFL           ;SEE IF PRINT EXPT-RCVD
3451  016350  001016                          BNE     LTG1T           ;IF NOT: BR
3452  016352  012704  021573                  MOV     #MSG12,R4
3453  016356  004737  017662                  JSR     PC,TTOUT        ;PRINT EXPT TAG
3454  016362  010103                          MOV     R1,R3
3455  016364  004737  020010                  JSR     PC,OCTP         ;PRINT EXPT
3456  016370  012704  021602                  MOV     #MSG13,R4
3457  016374  004737  017662                  JSR     PC,TTOUT        ;PRINT RCVD TAG
3458  016400  010203                          MOV     R2,R3
3459  016402  004737  020010                  JSR     PC,OCTP         ;PRINT RCVD
3460  016406  032777  004000  162154  LTG1T:  BIT     #4000,@SWR
3461  016414  001010                          BNE     LTG1C
3462  016416  012704  021644                  MOV     #MSG16,R4
3463  016422  004737  017662                  JSR     PC,TTOUT
3464  016426  013703  000676                  MOV     ITCNT,R3
3465  016432  004737  020010                  JSR     PC,OCTP         ;PRINT ITERATION
3466  016436  005237  000662          LTG1C:  INC     PFLG
3467  016442  000240                  LTG1X:  NOP
3468  016444  005777  162120                  TST     @SWR
3469  016450  100001                          BPL     LTG1X1          ;IF NOT STOP ON ERROR: BR
3470  016452  000000                          HALT
3471  016454  005737  000662          LTG1X1: TST     PFLG
3472  016460  001004                          BNE     LTG1XX          ;IF HAVE PRINTED: BR
3473  016462  032777  020000  162100          BIT     #20000,@SWR
3474  016470  001707                          BEQ     LTG1A
3475  016472  000240                  LTG1XX: NOP
3476  016474  005037  000732                  CLR     PEXFL           ;CLEAR EXPT-RCVD FLAG
3477  016500  000137  016504                  JMP     SCOPE           ;GO TO SCOPE
3478
3479
3480                                        ;SCOPE LOOP ON ERROR SUBROUTINE**********
3481
3482  016504  000240          SCOPE:  NOP
3483  016506  032777  040000  162054          BIT     #40000,@SWR     ;SEE IF LOOP ON ERROR
3484  016514  001001                          BNE     1$              ;IF SO: BR
3485  016516  000207                          RTS     PC              ;ELSE EXIT
3486  016520  000240                  1$:     NOP
3487  016522  005726                          TST     (SP)+           ;RESET STACK
3488  016524  000240                          NOP
3489  016526  000240                          NOP
3490  016530  000177  162152                  JMP     @SCOLP          ;LOOP ON ERROR
3491
```

```
3492                                        ;TEST ITERATION SUBROUTINE**********
3493
3494  016534  032777  004000  162026  ITER: BIT     #4000,@SWP        ;SEE IF ITERATIONS
3495  016542  001403                  BEQ     2$                ;IF SO: BR
3496  016544  005037  000676          1$:   CLR     ITCNT             ;CLEAR ITERATION COUNTER
3497  016550  000207                  RTS     PC                ;ELSE EXIT
3498  016552  005737  001014          2$:   TST     PCNTR             ;NO SUBTEST ITERATIONS ON FIRST PASS
3499  016556  001772                  BEQ     1$
3500  016560  005237  000676          INC     ITCNT             ;BUMP COUNTER
3501  016564  023737  000676  000602  CMP     ITCNT,ITAMT       ;SEE IF DONE ALL
3502  016572  001764                  BEQ     1$                ;IF SO: BR
3503  016574  005726                  TST     (SP)+             ;RESET STACK
3504  016576  017700  162106          MOV     @ITRLP,R0         ;SET ITERATION POINTER
3505  016602  000110                  JMP     (R0)              ;GO ITERATE
3506
3507                                        ;MANUAL INTERVENTION INHIBIT*********
3508
3509  016604  000240          INMT: NOP
3510  016606  012704  022435          MOV     #MSG43,R4
3511  016612  004737  017662          JSR     PC,TTOUT          ;GO PRINT INHIB MSG
3512  016616  000000                  HALT
3513  016620  000137  002460          JMP     TSCD2             ;RETURN TO SCHED
3514
3515
3516                                        ;NON-STANDARD MODE TEST HANDLER
3517
3518  016624  010046          NOST: MOV     R0,-(SP)          ;+SAVE R0
3519  016626  012700  000240          MOV     #240,R0           ;+SET UP INDEX
3520  016632  013760  001326  001056  MOV     TADX,TSTTBL(R0)   ;+
3521  016640  005720                  TST     (R0)+
3522  016642  012737  000047  001330  MOV     #47,TLAST         ;+SET END OF TEST
3523  016650  013760  001330  001056  MOV     TLAST,TSTTBL(R0)          ;+SET LAST TEST NUMBER
3524  016656  012600                  MOV     (SP)+,R0          ;+RESTORE R0
3525  016660  000207                  RTS     PC                ;+RETURN
3526
```

```
3527
3528                                        ;INITIALIZE SUBROUTINE**********
3529
3530  016662  000240                INIT1:  NOP
3531  016664  012777  000040  161626        MOV     #40,@CS         ;INIT
3532  016672  013777  000620  161620  INIT2: MOV     DRVN,@CS        ;SELECT DRIVE
3533  016700  013777  000660  161634        MOV     SLVN,@TC        ;SELECT SLAVE
3534  016706  000207                        RTS     PC              ;RETURN
3535
3536                                ;ROUTINES TO INITIALIZE SLAVE. THESE ROUTINES PLACE THE SLAVE
3537                                ;IN PROPER STATUS FOR THE CALLING TEST. INIT3 PLACES THE SLAVE IN
3538                                ;NRZ MODE AND OFF BOT; INIT4 PLACES THE SLAVE IN PE MODE AND OFF
3539                                ;BOT. IF THE SLAVE IS IN THE PROPER STATUS ON ENTRY NO ACTION IS TAKEN.
3540
3541                                ;SET SLAVE IN NRZ OFF BOT
3542  016710  013746  000772       INIT3:  MOV     UDES,-(SP)      ;SAVE TEST'S UNIT DESCRIPTION
3543  016714  012737  001400  000772        MOV     #1400,UDES      ;SET UNIT DESCRIPTION = NRZ
3544  016722  000410                        BR      INIT5           ;GO TO INIT5 ROUTINE
3545
3546                                ;SET SLAVE IN PE OFF BOT
3547  016724  013746  000772       INIT4:  MOV     UDES,-(SP)      ;SAVE TEST'S UNIT DESRIPTION
3548  016730  012737  002000  000772        MOV     #2000,UDES      ;SET UNIT DESRIPTION = PE
3549  016736  000402                        BR      INIT5           ;GO DO IT
3550
3551                                ;THIS ROUTINE IS ENTERED AT INIT WHEN THE CALLER HAS SETUP UDES.
3552                                ;IT IS ENTERED AT INIT5 WHEN EITHER INIT3 OR INIT4 HAS SET UP UDES.
3553  016740  013746  000772       INIT:   MOV     UDES,-(SP)      ;SAVE TEST'S UNIT DESRIPTION
3554  016744  012777  000040  161546  INIT5: MOV     #40,@CS         ;INIT CONTROLLER
3555  016752  013777  000620  161540        MOV     DRVN,@CS        ;SELECT TM03 DRIVE
3556  016760  013777  000660  161554        MOV     SLVN,@TC        ;SELECT TE16 SLAVE
3557  016766  013746  000772              MOV     UDES,-(SP)      ;GET SLAVE DESCRIPTION
3558  016772  042716  174377              BIC     #174377,(SP)    ;CLEAR ALL BUT DENSITY SELECT BITS
3559  016776  022726  001400              CMP     #1400,(SP)+     ;BRANCH IF REQUESTING PE MODE
3560  017002  001005                      BNE     1$
3561  017004  032777  000040  161510      BIT     #40,@DS         ;BRANCH IF SLAVE IS IN NRZ MODE
3562  017012  001420                      BEQ     4$              ;(PES = 0)
3563  017014  000404                      BR      2$
3564  017016  032777  000040  161476  1$: BIT     #40,@DS         ;BRANCH IF SLAVE IS IN PE MODE
3565  017024  001013                      BNE     4$
3566  017026  012777  000007  161454  2$: MOV     #7,@C1          ;REWIND SLAVE
3567  017034  032777  000200  161460  20$: BIT    #200,@DS        ;WAIT FOR READY
3568  017042  001774                      BEQ     20$
3569  017044  032777  020000  161450  3$: BIT     #20000,@DS      ;WAIT UNTIL PIP CLEARS
3570  017052  001374                      BNE     3$
3571  017054  053777  000772  161460  4$: BIS     UDES,@TC        ;LOAD SLAVE DESRIPTION
3572  017062  032777  000002  161432      BIT     #2,@DS          ;BRANCH IF NOT AT BOT
3573  017070  001407                      BEQ     6$
3574  017072  012777  000025  161410      MOV     #25,@C1         ;ERASE TO GET OFF BOT
3575  017100  032777  000200  161414  5$: BIT     #200,@DS        ;WAIT FOR READY
3576  017106  001774                      BEQ     5$
3577  017110  032777  000020  161404  6$: BIT     #20,@DS ;WAIT FOR SETTLEDOWN TO CLEAR
3578  017116  001374                      BNE     6$
3579  017120  012777  000011  161362      MOV     #11,@C1         ;RESET DRIVE
3580  017126  012637  000772              MOV     (SP)+,UDES      ;RESTORE UNIT DESRIPTION
3581  017132  000207                      RTS     PC              ;RETURN
3582
```

```
3583
3584
3585                                        ;MANUAL INSTRUCTION SUBROUTINE**********
3586
3587  017134  000240                INST:   NOP
3588  017136  004737  017662              JSR     PC,TTOUT        ;PRINT INSTRUCTION
3589  017142  012704  026740              MOV     #MMSG0,R4
3590  017146  004737  017662              JSR     PC,TTOUT        ;PRINT REPLY
3591  017152  012705  000674              MOV     #TEMP3,R5
3592  017156  012701  000001              MOV     #1,R1
3593  017162  012702  177777              MOV     #-1,R2
3594  017166  012703  000000              MOV     #0,R3
3595  017172  004737  017340              JSR     PC,TTR          ;AWAIT REPLY
3596  017176  000240                      NOP
3597  017200  000207                      RTS     PC              ;EXIT
3598
3599                                        ;MAG TAPE INTERRUPT HANDLER**********
3600
3601  017202  000240                MTINT:  NOP
3602  017204  013716  000664              MOV     RTRN,(SP)       ;SET RETURN FROM INTERUPT ADDRESS
3603  017210  000002                      RTI                     ;RETURN
3604
3605                                        ;TTY INTERRUPT HANDLER**********
3606
3607  017212  017746  161356        TTINT:  MOV     @TKB,-(SP)      ;GET CHARACTER
3608  017216  042716  000200              BIC     #200,(SP)       ;CLEAR PARITY BIT
3609  017222  122716  000003              CMPB    #3,(SP)         ;BRANCH IF NOT CONTROL C
3610  017226  001010                      BNE     1$
3611  017230  005737  001420              TST     CHNFLG          ;INHIBIT ^C IF IN CHAIN MODE
3612  017234  001005                      BNE     1$
3613  017236  005077  161324              CLR     @PSW            ;CLEAR PSW
3614  017242  000005                      RESET
3615  017244  000137  000200              JMP     @#200           ;RESTART
3616  017250  122716  000001        1$:   CMPB    #1,(SP)         ;BRANCH IF NOT ^A
3617  017254  001017                      BNE     2$
3618  017256  022737  000176  000570      CMP     #SWREG,SWR      ;BRANCH IF USING HARWARE SWR
3619  017264  001016                      BNE     3$
3620  017266  012737  177570  000570      MOV     #177570,SWR     ;INVOKE HARDWARE SWR
3621  017274  004737  020534              JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
3622  017300  012704  023432              MOV     #MSG63,R4       ;TYPE 'HARDWARE SWR IN USE'
3623  017304  004737  017662              JSR     PC,TTOUT
3624  017310  004737  020556              JSR     PC,.RESTORE
3625  017314  122716  000007        2$:   CMPB    #7,(SP)         ;BRANCH IF NOT ^G
3626  017320  001005                      BNE     4$
3627  017322  012737  000176  000570  3$: MOV     #SWREG,SWR      ;INVOKE SOFTWARE SWR
3628  017330  004737  020436              JSR     PC,GTSWR        ;GET SWITCHES
3629  017334  005726                4$:   TST     (SP)+           ;POP CHARACTER OFF STACK
3630  017336  000002                      RTI                     ;RETURN
3631
```

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

017340  010146              TTR:    MOV     R1,-(SP)        ;SAVE CHARACTER COUNT
017342  011601              10$:    MOV     (SP),R1         ;RESTORE CHARACTER COUNT (FOR ^U)
017344  005037  000670              CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
017350  005000                      CLR     R0
017352  004737  017620      1$:     JSR     PC,TTIN         ;GO READ CHARACTER
017356  122737  000003  000612      CMPB    #3,TIB          ;BRANCH IF NOT ^C
017364  001003                      BNE     11$
017366  000005                      RESET                   ;RESET
017370  000137  000200              JMP     @#200           ;RESTART PROGRAM
017374  122737  000015  000612  11$:CMPB    #15,TIB         ;SEE IF CR
017402  001004                      BNE     2$              ;IF NOT: BR
017404  005737  000670              TST     TEMP1           ;SEE IF FIRST CHARACTER
017410  001471                      BEQ     9$              ;IF SO: BR
017412  000457                      BR      6$
017414  122737  000025  000612  2$: CMPB    #25,TIB         ;BRANCH IF NOT CONTROL U
017422  001005                      BNE     21$
017424  012704  023360              MOV     #MSG59,R4       ;TYPE <CR><LF>
017430  004737  017662              JSR     PC,TTOUT
017434  000742                      BR      10$
017436  122737  000177  000612  21$:CMPB    #177,TIB        ;BRANCH IF NOT 'RUBOUT'
017444  001012                      BNE     3$
017446  000241                      CLC
017450  006000                      ROR     R0              ;REMOVE LAST CHAR
017452  006200                      ASR     R0
017454  006200                      ASR     R0
017456  012704  023362              MOV     #MSG60,R4       ;TYPE '\'
017462  004737  017662              JSR     PC,TTOUT
017466  005201                      INC     R1              ;DECREMENT CHARS RECEIVED COUNT
017470  000730                      BR      1$
017472  122737  000060  000612  3$: CMPB    #60,TIB         ;SEE IF CHAR IS LESS THAN 0
017500  101402                      BLOS    4$              ;IF NOT: BR
017502  000137  017600              JMP     TIMER           ;ELSE GO TO ERROR
017506  122737  000070  000612  4$: CMPB    #70,TIB         ;SEE IF CHAR IS GREATER THAN 7
017514  101002                      BHI     5$              ;IF NOT: BR
017516  000137  017600              JMP     TIMER           ;ELSE GO TO ERROR
017522  005237  000670          5$: INC     TEMP1           ;SET FIRST CHARACTER FLAG
017526  006300                      ASL     R0
017530  006300                      ASL     R0              ;SHIFT 3 LEFT
017532  006300                      ASL     R0
```

```
3688  017534  042737  177770  000612          BIC     #177770,TIB     ;STRIP ASCII
3689  017542  053700  000612                  BIS     TIB,R0          ;LOAD CHARACTER
3690  017546  005301                          DEC     R1              ;SEE IF DONE
3691  017550  001300                          BNE     1$              ;IF NOT: BR
3692  017552  020002                  6$:     CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
3693  017554  101402                          BLOS    7$              ;IF NOT: BR
3694  017556  000137  017600                  JMP     TINER           ;ELSE GO TO ERROR
3695  017562  020300                  7$:     CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
3696  017564  101402                          BLOS    8$              ;IF NOT: BR
3697  017566  000137  017600                  JMP     TINER           ;ELSE GO TO ERROR
3698  017572  010015                  8$:     MOV     R0,(R5)         ;LOAD VALUE
3699  017574  005726                  9$:     TST     (SP)+           ;POP CHAR COUNT OFF STACK
3700  017576  000207                          RTS     PC              ;EXIT
3701
```

```
3702
3703                                        ;TTY ENTRY ERROR SUBROUTINE**********
3704
3705  017600  012704  022375          TINER:  MOV     #MSG40,R4
3706  017604  004737  017662                  JSR     PC,TTOUT        ;PRINT?
3707  017610  005726                          TST     (SP)+           ;POP CHAR COUNT OFF STACK
3708  017612  162716  000020                  SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
3709  017616  000207                          RTS     PC              ;REDO VALUE ENTRY
3710
3711                                        ;TTY READ SUBROUTINE**********
3712
3713  017620  005277  160746          TTIN:   INC     @TKS
3714  017624  105777  160742          1$:     TSTB    @TKS
3715  017630  100375                          BPL     1$
3716  017632  017737  160736  000612          MOV     @TKB,TIB
3717  017640  042737  000200  000612          BIC     #200,TIB                ;STRIP PARITY BIT
3718  017646  013737  000612  000610          MOV     TIB,TOB                 ;MOVE CHAR TO TTY OUTPUT BFR
3719  017654  004737  017762                  JSR     PC,TOG                  ;AND ECHO IT
3720  017660  000207                          RTS     PC
3721
3722                                        ;TTY OUTPUT SUBROUTINE***********
3723
3724  017662  112437  000610          TTOUT:  MOVB    (R4)+,TOB
3725  017666  122737  000043  000610          CMPB    #43,TOB
3726  017674  001440                          BEQ     TEX
3727  017676  122737  000045  000610          CMPB    #45,TOB
3728  017704  001403                          BEQ     1$
3729  017706  004737  017762                  JSR     PC,TOG
3730  017712  000763                          BR      TTOUT
3731  017714  112737  000015  000610  1$:     MOVB    #15,TOB
3732  017722  004737  017762                  JSR     PC,TOG
3733  017726  012703  000004                  MOV     #4,R3
3734  017732  005037  000610          2$:     CLR     TOB
3735  017736  004737  017762                  JSR     PC,TOG
3736  017742  005303                          DEC     R3
3737  017744  001372                          BNE     2$                      ;DO FILLERS
3738  017746  112737  000012  000610          MOVB    #12,TOB
3739  017754  004737  017762                  JSR     PC,TOG
3740  017760  000740                          BR      TTOUT
3741  017762  105777  160610          TOG:    TSTB    @TPS
3742  017766  100375                          BPL     TOG
3743  017770  113777  000610  160602          MOVB    TOB,@TPB
3744  017776  000207                  TEX:    RTS     PC
3745
3746
3747                                        ;OCTAL OUTPUT SUBROUTINE***********
3748
3749  020000  012737  000001  020230  OCTPE:  MOV     #1,OFL
3750  020006  000402                          BR      OCTPE1
3751  020010  005037  020230          OCTP:   CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
3752  020014  010304                  OCTPE1: MOV     R3,R4           ;SEE IF NUMBER IS ZERO
3753  020016  001006                          BNE     OCTP0           ;IF NOT ZERO: BR
3754  020020  005737  020230                  TST     OFL             ;SEE IF PRINT ALL 0
3755  020024  001003                          BNE     OCTP0           ;IF SO: BR
3756  020026  004737  020210                  JSR     PC,OCTPG1       ;ELSE PRINT ZERO
3757  020032  000447                          BR      OCTP3           ;SPACE AND EXIT
```

```
3758  020034  032704  100000          OCTP0:  BIT     #100000,R4      ;SEE IF MSD = 1
3759  020040  001405                          BEQ     OCTP1           ;IF NOT: BR
3760  020042  012704  000001                  MOV     #1,R4
3761  020046  004737  020166                  JSR     PC,OCTPG        ;PRINT 1
3762  020052  000403                          BR      OCTP2
3763  020054  005004                  OCTP1:  CLR     R4
3764  020056  004737  020166                  JSR     PC,OCTPG        ;PRINT 0
3765  020062  010304                  OCTP2:  MOV     R3,R4
3766  020064  006004                          ROR     R4
3767  020066  006004                          ROR     R4
3768  020070  006004                          ROR     R4              ;POSITION DIGIT
3769  020072  006004                          ROR     R4
3770  020074  000304                          SWAB    R4
3771  020076  004737  020166                  JSR     PC,OCTPG        ;PRINT DIGIT 2
3772  020102  010304                          MOV     R3,R4
3773  020104  006004                          ROR     R4
3774  020106  000304                          SWAB    R4
3775  020110  004737  020166                  JSR     PC,OCTPG        ;PRINT DIGIT 3
3776  020114  010304                          MOV     R3,R4
3777  020116  006104                          ROL     R4
3778  020120  006104                          ROL     R4
3779  020122  000304                          SWAB    R4
3780  020124  004737  020166                  JSR     PC,OCTPG        ;PRINT DIGIT 4
3781  020130  010304                          MOV     R3,R4
3782  020132  006004                          ROR     R4
3783  020134  006004                          ROR     R4
3784  020136  006004                          ROR     R4
3785  020140  004737  020166                  JSR     PC,OCTPG
3786  020144  010304                          MOV     R3,R4
3787  020146  004737  020166                  JSR     PC,OCTPG        ;PRINT DIGIT 5
3788  020152  012737  000240  000610  OCTP3:  MOV     #240,TOB
3789  020160  004737  017762                  JSR     PC,TOG          ;PRINT SPACE
3790  020164  000207                          RTS     PC              ;EXIT
3791
3792  020166  042704  177770          OCTPG:  BIC     #177770,R4
3793  020172  001004                          BNE     OCTPG0
3794  020174  005737  020230                  TST     OFL
3795  020200  001001                          BNE     OCTPG0
3796  020202  000207                          RTS     PC
3797
3798  020204  005237  020230          OCTPG0: INC     OFL
3799  020210  052704  000260          OCTPG1: BIS     #260,R4
3800  020214  010437  000610                  MOV     R4,TOB
3801  020220  004737  017762                  JSR     PC,TOG
3802  020224  010304                          MOV     R3,R4
3803  020226  000207                          RTS     PC
3804  020230  000000                  OFL:    0                       ;FIRST CHAR FLAG
3805
```

```
3806
3807                                        ;DATA CHARACTER OUTPUT SUBROUTINE**********
3808
3809  020232  012704  000010        DOUT:   MOV     #10,R4          ;SET NUMBER TO PRINT
3810  020236  110337  000610                MOVB    R3,TOB
3811  020242  105777  160330        1$:     TSTB    @TPS
3812  020246  100375                BPL     1$
3813  020250  132737  000200  000610       BITB    #200,TOB
3814  020256  001404                BEQ     2$
3815  020260  012777  000061  160312       MOV     #061,@TPB
3816  020266  000403                BR      3$
3817  020270  012777  000060  160302  2$:  MOV     #060,@TPB
3818  020276  006337  000610        3$:     ASL     TOB
3819  020302  005304                DEC     R4
3820  020304  001356                BNE     1$
3821  020306  000207                RTS     PC
3822
3823  020310  013703  000674        DOUTD:  MOV     TEMP3,R3
3824  020314  000303                SWAB    R3
3825  020316  004737  020232              JSR     PC,DOUT
3826  020322  013703  000674              MOV     TEMP3,R3
3827  020326  004737  020232              JSR     PC,DOUT
3828  020332  000207                RTS     PC
3829
3830                                        ;TE16 SERIAL NUMBER PRINT SUBROUTINE**********
3831
3832  020334  010304                SNPT:   MOV     R3,R4
3833  020336  000304                SWAB    R4
3834  020340  006004                ROR     R4
3835  020342  006004                ROR     R4
3836  020344  006004                ROR     R4
3837  020346  006004                ROR     R4              ;GET FIRST DIGIT
3838  020350  004737  020412        JSR     PC,SNPG         ;PRINT
3839  020354  010304                MOV     R3,R4
3840  020356  000304                SWAB    R4              ;GET SECOND DIGIT
3841  020360  004737  020412        JSR     PC,SNPG         ;PRINT
3842  020364  010304                MOV     R3,R4
3843  020366  006004                ROR     R4
3844  020370  006004                ROR     R4
3845  020372  006004                ROR     R4
3846  020374  006004                ROR     R4
3847  020376  004737  020412        JSR     PC,SNPG         ;PRINT THIRD DIGIT
3848  020402  010304                MOV     R3,R4
3849  020404  004737  020412        JSR     PC,SNPG         ;PRINT FOURTH DIGIT
3850  020410  000207                RTS     PC              ;EXIT
3851  020412  012737  000260  000610  SNPG:  MOV     #260,TOB        ;SET BASE = 0
3852  020420  042704  177760              BIC     #177760,R4      ;MASK DIGIT
3853  020424  050437  000610              BIS     R4,TOB          ;SET ASCII
3854  020430  004737  017762              JSR     PC,TOG          ;TYPE DIGIT
3855  020434  000207                RTS     PC              ;RETURN
```

```
3856
3857                                    ;ROUTINE TO LOAD CONTENTS OF SOFTWARE SWITCH REGISTER.
3858                                    ;IF A CONTROL G (^G) IS TYPED THE SOFTWARE SWITCH REGISTER IS LOADED
3859  020436  022737  000176  000570    GTSWR:  CMP     #SWREG,SWR      ;BRANCH IF SOFTWARE SWR
3860  020444  001032                            BNE     1$              ;NOT INVOKED
3861  020446  004737  020534                    JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
3862  020452  012704  027265                    MOV     #$MSWR,R4       ;TYPE 'SWR = '
3863  020456  004737  017662                    JSR     PC,TTOUT
3864  020462  017703  160102                    MOV     @SWR,R3         ;GET CURRENT VALUE
3865  020466  004737  020000                    JSR     PC,OCTPE        ;AND TYPE IT
3866  020472  012704  027275                    MOV     #$MNEW,R4       ;ASK FOR NEW VALUE
3867  020476  004737  017662                    JSR     PC,TTOUT
3868  020502  013705  000570                    MOV     SWR,R5          ;NEW VALUE WILL BE RETURNED IN (R5)
3869  020506  012701  000007                    MOV     #7,R1           ;LIMIT TO 7 CHARACTERS
3870  020512  012702  177777                    MOV     #177777,R2      ;LIMIT RESPONSE TO BETWEEN
3871  020516  012703  000000                    MOV     #0,R3           ;0 AND 177777
3872  020522  004737  017340                    JSR     PC,TTR          ;GET RESPONSE
3873  020526  004737  020556                    JSR     PC,.RESTORE     ;RESTORE REGISTERS
3874  020532  000207                    1$:     RTS     PC              ;RETURN TO CALLER
3875                                    ;;ROUTINE TO SAVE REGISTERS ON THE STACK
3876  020534  010546                    .SAVE:  MOV     %5,-(SP)        ;;R5 IS SAVED AT 12(SP)
3877  020536  010446                            MOV     %4,-(SP)        ;;R4 IS SAVED AT 10(SP)
3878  020540  010346                            MOV     %3,-(SP)        ;;R3 IS SAVED AT 6(SP)
3879  020542  010246                            MOV     %2,-(SP)        ;;R2 IS SAVED AT 4(SP)
3880  020544  010146                            MOV     %1,-(SP)        ;;R1 IS SAVED AT 2(SP)
3881  020546  010046                            MOV     %0,-(SP)        ;;R0 IS SAVED AT (SP)
3882  020550  016646  000014                    MOV     14(SP),-(SP)    ;;PUSH RETURN PC ON THE STACK
3883  020554  000207                            RTS     PC              ;;RETURN TO CALLER
3884
3885                                    ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
3886  020556  012666  000014            .RESTORE:MOV    (SP)+,14(SP)    ;;STORE RETURN PC ON STACK
3887  020562  012600                            MOV     (SP)+,%0
3888  020564  012601                            MOV     (SP)+,%1
3889  020566  012602                            MOV     (SP)+,%2
3890  020570  012603                            MOV     (SP)+,%3
3891  020572  012604                            MOV     (SP)+,%4
3892  020574  012605                            MOV     (SP)+,%5
3893  020576  000207                            RTS     PC              ;;RETURN
3894
3895
3896                                    ;MESSAGE TABLE**********
3897
3898  020600  022445  046524  031460    MSG1:   .ASCII  '%%TM03-TE16/TU77 CONTROL LOGIC TEST- PART I (CZTEAB0)';++B
3899  020606  052055  030505  027466
3900  020614  052524  033467  041440
3901  020622  047117  051124  046117
3902  020630  046040  043517  041511
3903  020636  052040  051505  026524
3904  020644  050040  051101  020124
3905  020652  020111  041450  052132
3906  020660  040505  030102     051
3907  020665     045  025052  040452            .ASCII  /%***ASSURE TAPE IS AT BOT***/
3908  020672  051523  051125  020105
3909  020700  040524  042520  044440
3910  020706  020123  052101  041040
3911  020714  052117  025052     052
```

```
 3912  020721     045  054524  042520          .ASCII  /%TYPE <CR> TO TERMINATE RESPONSE & ^C TO RESTART%#/
 3913  020726  036040  051103  020076
 3914  020734  047524  052040  051105
 3915  020742  044515  040516  042524
 3916  020750  051040  051505  047520
 3917  020756  051516  020105  020046
 3918  020764  041536  052040  020117
 3919  020772  042522  052123  051101
 3920  021000  022524     043
 3921  021003     045  051104  053111  MSG2:   .ASCII  /%DRIVE NUMBER OR (CR) WHEN DONE #/
 3922  021010  020105  052516  041115
 3923  021016  051105  047440  020122
 3924  021024  041450  024522  053440
 3925  021032  042510  020116  047504
 3926  021040  042516  021440
 3927  021044  022445  047506  020122  MSG2A:  .ASCII  /%%FOR DRIVE ADDRESS TEST;/
 3928  021052  051104  053111  020105
 3929  021060  042101  051104  051505
 3930  021066  020123  042524  052123
 3931  021074     073
 3932  021075     045  042440  052116          .ASCII  /% ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
 3933  021102  051105  042440  050130
 3934  021110  020124  051104  053111
 3935  021116  020105  052516  041115
 3936  021124  051105  020054  046101
 3937  021132  020114  052117  042510
 3938  021140  051522  051440  047510
 3939  021146  046125  020104  042502
 3940  021154  047040  047117  042455
 3941  021162  044530  052123  047101
 3942  021170  027124     043
 3943  021173     045  047516  026516  MSG3:   .ASCII  /%NON-EXIST DRIVE #/
 3944  021200  054105  051511  020124
 3945  021206  051104  053111  020105
 3946  021214     043
 3947  021215     045  044122  042040  MSG4:   .ASCII  /%RH DETECTED #/
 3948  021222  052105  041505  042524
 3949  021230  020104     043
 3950  021233     045  046524  031460  MSG5:   .ASCII  /%TM03 DETECTED #/
 3951  021240  042040  052105  041505
 3952  021246  042524  020104     043
 3953  021253     105  050130  026524  MSG6:   .ASCII  /EXPT-NOT RECVD#/
 3954  021260  047516  020124  042522
 3955  021266  053103  021504
 3956  021272  041522  042126  047055  MSG7:   .ASCII  /RCVD-NOT EXPT#/
 3957  021300  052117  042440  050130
 3958  021306  021524
 3959  021310  051445  040514  042526  MSG8:   .ASCII  /%SLAVE NUMBER OR (CR) WHEN DONE #/
 3960  021316  047040  046525  042502
 3961  021324  020122  051117  024040
 3962  021332  051103  020051  044127
 3963  021340  047105  042040  047117
 3964  021346  020105     043
 3965  021351     045  043045  051117  MSG8A:  .ASCII  /%%FOR SLAVE ADDRESS TEST;/
 3966  021356  051440  040514  042526
 3967  021364  040440  042104  042522
```

```
 3968  021372  051523  052040  051505
 3969  021400  035524
 3970  021402  020045  047105  042524          .ASCII  /% ENTER EXPT SLAVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
 3971  021410  020122  054105  052120
 3972  021416  051440  040514  042526
 3973  021424  047040  046525  042502
 3974  021432  026122  040440  046114
 3975  021440  047440  044124  051105
 3976  021446  020123  044123  052517
 3977  021454  042114  041040  020105
 3978  021462  047516  026516  054105
 3979  021470  051511  040524  052116
 3980  021476  021456
 3981  021500  047045  047117  042455  MSG9:   .ASCII  /%NON-EXIST SLAVE #/
 3982  021506  044530  052123  051440
 3983  021514  040514  042526  021440
 3984  021522  051045  040505  020104  MSG10:  .ASCII  /%READ CONT BUS PAR #/
 3985  021530  047503  052116  041040
 3986  021536  051525  050040  051101
 3987  021544  021440
 3988  021546  053445  044522  042524  MSG11:  .ASCII  /%WRITE CONT BUS PAR #/
 3989  021554  041440  047117  020124
 3990  021562  052502  020123  040520
 3991  021570  020122     043
 3992  021573     040  054105  052120  MSG12:  .ASCII  / EXPT #/
 3993  021600  021440
 3994  021602  051040  053103  020104  MSG13:  .ASCII  / RCVD #/
 3995  021610     043
 3996  021611     045  051115  041040  MSG14:  .ASCII  /%MR BITS 4-0#/
 3997  021616  052111  020123  026464
 3998  021624  021460
 3999  021626  046445  020122  044502  MSG15:  .ASCII  /%MR BITS 15-7#/
 4000  021634  051524  030440  026465
 4001  021642  021467
 4002  021644  044445  042524  035122  MSG16:  .ASCII  /%ITER: #/
 4003  021652  021440
 4004  021654  052045  020103  044502  MSG18:  .ASCII  /%TC BITS 12-0 #/
 4005  021662  051524  030440  026462
 4006  021670  020060     043
 4007  021673     045  041506  041040  MSG19:  .ASCII  /%FC BITS 15-0 #/
 4008  021700  052111  020123  032461
 4009  021706  030055  021440
 4010  021712  043045  047125  041440  MSG20:  .ASCII  /%FUN CODE BITS 5-1 OF C1 #/
 4011  021720  042117  020105  044502
 4012  021726  051524  032440  030455
 4013  021734  047440  020106  030503
 4014  021742  021440
 4015  021744  043445  020117  044502  MSG21:  .ASCII  /%GO BIT NOT CORRECT AT START #/
 4016  021752  020124  047516  020124
 4017  021760  047503  051122  041505
 4018  021766  020124  052101  051440
 4019  021774  040524  052122  021440
 4020  022002  043445  020117  044502  MSG22:  .ASCII  /%GO BIT NOT SET #/
 4021  022010  020124  047516  020124
 4022  022016  042523  020124     043
 4023  022023     045  047507  041040  MSG23:  .ASCII  /%GO BIT NOT RESET BY INIT #/
```

```
 4024  022030  052111  047040  052117
 4025  022036  051040  051505  052105
 4026  022044  041040  020131  047111
 4027  022052  052111  021440
 4028  022056  042045  054522  047040  MSG24:  .ASCII  /%DRY NOT SET BY INIT #/
 4029  022064  052117  051440  052105
 4030  022072  041040  020131  047111
 4031  022100  052111  021440
 4032  022104  042045  054522  047040  MSG25:  .ASCII  /%DRY NOT RESET BY GO=1#/
 4033  022112  052117  051040  051505
 4034  022120  052105  041040  020131
 4035  022126  047507  030475     043
 4036  022133     045  051104  020131  MSG25A: .ASCII  /%DRY NOT SET BY GO=0#/
 4037  022140  047516  020124  042523
 4038  022146  020124  054502  043440
 4039  022154  036517  021460
 4040  022160  047045  020117  047111  MSG26:  .ASCII  /%NO INTERRUPT RETURNED#/
 4041  022166  042524  051122  050125
 4042  022174  020124  042522  052524
 4043  022202  047122  042105     043
 4044  022207     045  040502  020104  MSG27:  .ASCII  /%BAD STATUS#/
 4045  022214  052123  052101  051525
 4046  022222     043
 4047  022223     040  047123  020072  MSG30:  .ASCII  / SN: #/
 4048  022230     043
 4049  022231     045  051105  020122  MSG31:  .ASCII  /%ERR NOT SET #/
 4050  022236  047516  020124  042523
 4051  022244  020124     043
 4052  022247     045  052101  020101  MSG32:  .ASCII  /%ATA NOT SET #/
 4053  022254  047516  020124  042523
 4054  022262  020124     043
 4055  022265     045  051501  041040  MSG33:  .ASCII  /%AS BIT NOT SET #/
 4056  022272  052111  047040  052117
 4057  022300  051440  052105  021440
 4058  022306  051445  020103  047516  MSG34:  .ASCII  /%SC NOT SET #/
 4059  022314  020124  042523  020124
 4060  022322     043
 4061  022323     045  051124  020105  MSG35:  .ASCII  /%TRE NOT SET #/
 4062  022330  047516  020124  042523
 4063  022336  020124     043
 4064  022341     045  046123  020101  MSG36:  .ASCII  /%SLA NOT SET #/
 4065  022346  047516  020124  042523
 4066  022354  020124     043
 4067  022357     045  051523  020103  MSG37:  .ASCII  /%SSC NOT SET #/
 4068  022364  047516  020124  042523
 4069  022372  020124     043
 4070  022375     040  020077     043  MSG40:  .ASCII  / ? #/
 4071  022401     045  042445  042116  MSG41:  .ASCII  /%%END OF PASS #/
 4072  022406  047440  020106  040520
 4073  022414  051523  021440
 4074  022420  042045  040505  020104  MSG42:  .ASCII  /%DEAD TRACK #/
 4075  022426  051124  041501  020113
 4076  022434     043
 4077  022435     045  046445  047101  MSG43:  .ASCII  /%%MANUAL TESTS (14-17) INHIBITED: HALT%/
 4078  022442  040525  020114  042524
 4079  022450  052123  020123  030450
```

```
 4080  022456  026464  033461  020051
 4081  022464  047111  044510  044502
 4082  022472  042524  035104  044040
 4083  022500  046101  022524
 4084  022504  042522  042523  042514          .ASCII  /RESELECT AND PRESS CONTINUE%#/
 4085  022512  052103  040440  042116
 4086  022520  050040  042522  051523
 4087  022526  041440  047117  044524
 4088  022534  052516  022505     043
 4089  022541     045  042522  044507  MSG44:  .ASCII  /%REGISTER START: #/
 4090  022546  052123  051105  051440
 4091  022554  040524  052122  020072
 4092  022562     043
 4093  022563     045  042526  052103  MSG45:  .ASCII  /%VECTOR ADDRESS: #/
 4094  022570  051117  040440  042104
 4095  022576  042522  051523  020072
 4096  022604     043
 4097  022605     045  051503  020061  MSG46:  .ASCII  /%CS1    WC     BA     FC     CS2    DS     ER     AS/
 4098  022612  020040  053440  020103
 4099  022620  020040  020040  040502
 4100  022626  020040  020040  043040
 4101  022634  020103  020040  020040
 4102  022642  051503  020062  020040
 4103  022650  042040  020123  020040
 4104  022656  020040  051105  020040
 4105  022664  020040  040440     123
 4106  022671     040  020040  020040          .ASCII  /     MR     TC%#/
 4107  022676  051115  020040  020040
 4108  022704  052040  022503     043
 4109  022711     045  047516  020124  MSG47:  .ASCII  /%NOT RESET BY DRIVE CLEAR#/
 4110  022716  042522  042523  020124
 4111  022724  054502  042040  044522
 4112  022732  042526  041440  042514
 4113  022740  051101     043
 4114  022743     045  046101  044120  MSG50:  .ASCII  /%ALPHA NOT SET#/
 4115  022750  020101  047516  020124
 4116  022756  042523  021524
 4117  022762  052445  042516  050130  MSG51:  .ASCII  /%UNEXPECTED ERROR BITS#/
 4118  022770  041505  042524  020104
 4119  022776  051105  047522  020122
 4120  023004  044502  051524     043
 4121  023011     045  040502  020104  MSG53:  .ASCII  /%BAD LRC #/
 4122  023016  051114  020103     043
 4123  023023     045  040502  020104  MSG54:  .ASCII  /%BAD CK #/
 4124  023030  045503  021440
 4125  023034  051445  052105  050125  MSG55:  .ASCII  /%SETUP ERROR: CHECK WRAP 0 WITH CONTROL LOGIC TEST II TEST 7#/
 4126  023042  042440  051122  051117
 4127  023050  020072  044103  041505
 4128  023056  020113  051127  050101
 4129  023064  030040  053440  052111
 4130  023072  020110  047503  052116
 4131  023100  047522  020114  047514
 4132  023106  044507  020103  042524
 4133  023114  052123  044440  020111
 4134  023122  042524  052123  033440
 4135  023130     043
```

```
 4136  023131     045  052123  052101  MSG56:  .ASCII  /%STATIC TESTS ONLY: #/
 4137  023136  041511  052040  051505
 4138  023144  051524  047440  046116
 4139  023152  035131  021440
 4140  023156  052045  047515  020063  MSG57:  .ASCII  /%TM03 DRIVE: #/
 4141  023164  051104  053111  035105
 4142  023172  021440
 4143  023174  044445  020123  047503  MS57A:  .ASCII  /%IS CONTROLLER JUMPERED IN NON-STANDARD MODE,/<15><12>
 4144  023202  052116  047522  046114
 4145  023210  051105  045040  046525
 4146  023216  042520  042522  020104
 4147  023224  047111  047040  047117
 4148  023232  051455  040524  042116
 4149  023240  051101  020104  047515
 4150  023246  042504  006454     012
 4151  023253     124  050131  020105          .ASCII  /TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD ?     #/
 4152  023260  020062  047506  020122
 4153  023266  047516  036516  052123
 4154  023274  047101  040504  042122
 4155  023302  047440  020122  051103
 4156  023310  043040  051117  051440
 4157  023316  040524  042116  051101
 4158  023324  020104  020077  020040
 4159  023332  020040     043
 4160  023335     045  042524  033061  MSG58:  .ASCII  '%TE16/TU77 SLAVE: #';++B
 4161  023342  052057  033525  020067
 4162  023350  046123  053101  035105
 4163  023356  021440
 4164  023360  021445                  MSG59:  .ASCII  /%#/
 4165  023362  021534                  MSG60:  .ASCII  /\#/
 4166  023364  051045  046505  053117  MSG62:  .ASCII  /%REMOVE TMDP FROM SLAVE TO BE TESTED%#/
 4167  023372  020105  046524  050104
 4168  023400  043040  047522  020115
 4169  023406  046123  053101  020105
 4170  023414  047524  041040  020105
 4171  023422  042524  052123  042105
 4172  023430  021445
 4173  023432  044045  051101  053504  MSG63:  .ASCII  /%HARDWARE SWR IN USE%#/
 4174  023440  051101  020105  053523
 4175  023446  020122  047111  052440
 4176  023454  042523  021445
 4177  023460  051445  040514  042526  MSG64:  .ASCII  /%SLAVE TYPE: #/ ;++B
 4178  023466  052040  050131  035105
 4179  023474  021440
 4180  023476  052524  033467     043  MSG65:  .ASCII  /TU77#/ ;++B
 4181  023503     124  030505  021466  MSG66:  .ASCII  /TE16#/ ;++B
 4182  023510  051445  040514  042526  MSG67:  .ASCII  /%SLAVE TYPE (0=TE16,1=TU77): #/ ;++B
 4183  023516  052040  050131  020105
 4184  023524  030050  052075  030505
 4185  023532  026066  036461  052524
 4186  023540  033467  035051  021440
 4187  023546  022445  047111  047503  MSG68:  .ASCII  /%%INCORRECT SLAVE TYPE:!! PROGRAM ABORTED#/ ;++B
 4188  023554  051122  041505  020124
 4189  023562  046123  053101  020105
 4190  023570  054524  042520  020441
 4191  023576  020041  051120  043517
```

```
4192  023604  040522  020115  041101
4193  023612  051117  042524  021504
4194  023620  046111  042514  040507  MSG69:  .ASCII  /ILLEGAL#/      ;++B
4195  023626  021514
4196                                          ;TEST HEADER**********
4197
4198  023630  022445  047514  044507  MSLT1:  .ASCII  /%%LOGIC TEST 1: DRIVE ADDRESSING (M8909 RH)#/
4199  023636  020103  042524  052123
4200  023644  030440  020072  051104
4201  023652  053111  020105  042101
4202  023660  051104  051505  044523
4203  023666  043516  024040  034115
4204  023674  030071  020071  044122
4205  023702  021451
4206  023704  022445  047514  044507  MSLT2:  .ASCII  /%%LOGIC TEST 2: REGISTER ADDRESSING (M8909 RH)#/
4207  023712  020103  042524  052123
4208  023720  031040  020072  042522
4209  023726  044507  052123  051105
4210  023734  040440  042104  042522
4211  023742  051523  047111  020107
4212  023750  046450  034470  034460
4213  023756  051040  024510     043
4214  023763     045  046045  043517  MSLT3:  .ASCII  /%%LOGIC TEST 3: CONTROL BUS TEST (RH M8905-YB M8909)#/
4215  023770  041511  052040  051505
4216  023776  020124  035063  041440
4217  024004  047117  051124  046117
4218  024012  041040  051525  052040
4219  024020  051505  020124  051050
4220  024026  020110  034115  030071
4221  024034  026465  041131  046440
4222  024042  034470  034460  021451
4223  024050  022445  047514  044507  MSLT4:  .ASCII  /%%LOGIC TEST 4: SLAVE ADDRESSING (M8905-YB M8933)#/
4224  024056  020103  042524  052123
4225  024064  032040  020072  046123
4226  024072  053101  020105  042101
4227  024100  051104  051505  044523
4228  024106  043516  024040  034115
4229  024114  030071  026465  041131
4230  024122  046440  034470  031463
4231  024130  021451
4232  024132  022445  047514  044507  MSLT5:  .ASCII  /%%LOGIC TEST 5: MR BIT TEST (M8905-YB)#/
4233  024140  020103  042524  052123
4234  024146  032440  020072  051115
4235  024154  041040  052111  052040
4236  024162  051505  020124  046450
4237  024170  034470  032460  054455
4238  024176  024502     043
4239  024201     045  046045  043517  MSLT6:  .ASCII  /%%LOGIC TEST 6: TC BIT TEST (M8905-YB)#/
4240  024206  041511  052040  051505
4241  024214  020124  035066  052040
4242  024222  020103  044502  020124
4243  024230  042524  052123  024040
4244  024236  034115  030071  026465
4245  024244  041131  021451
4246  024250  022445  047514  044507  MSLT7:  .ASCII  /%%LOGIC TEST 7: FC BIT TEST (M8905-YB)#/
4247  024256  020103  042524  052123
```

```
 4248  024264  033440  020072  041506
 4249  024272  041040  052111  052040
 4250  024300  051505  020124  046450
 4251  024306  034470  032460  054455
 4252  024314  024502     043
 4253  024317     045  046045  043517  MSLT10: .ASCII  /%%LOGIC TEST 10: FUNCTION BIT TEST (M8905-YB)#/
 4254  024324  041511  052040  051505
 4255  024332  020124  030061  020072
 4256  024340  052506  041516  044524
 4257  024346  047117  041040  052111
 4258  024354  052040  051505  020124
 4259  024362  046450  034470  032460
 4260  024370  054455  024502     043
 4261  024375     045  046045  043517  MSLT11: .ASCII  /%%LOGIC TEST 11: GO BIT TEST (M8909)#/
 4262  024402  041511  052040  051505
 4263  024410  020124  030461  020072
 4264  024416  047507  041040  052111
 4265  024424  052040  051505  020124
 4266  024432  046450  034470  034460
 4267  024440  021451
 4268  024442  022445  047514  044507  MSLT12: .ASCII  /%%LOGIC TEST 12: DRIVE READY BIT (M8909)#/
 4269  024450  020103  042524  052123
 4270  024456  030440  035062  042040
 4271  024464  044522  042526  051040
 4272  024472  040505  054504  041040
 4273  024500  052111  024040  034115
 4274  024506  030071  024471     043
 4275  024513     045  046045  043517  MSLT13: .ASCII  /%%LOGIC TEST 13: INTERRUPT TEST (RH)#/
 4276  024520  041511  052040  051505
 4277  024526  020124  031461  020072
 4278  024534  047111  042524  051122
 4279  024542  050125  020124  042524
 4280  024550  052123  024040  044122
 4281  024556  021451
 4282  024560  022445  047514  044507  MSLT14: .ASCII  /%%LOGIC TEST 14: STATUS AT BOT,ON-LINE,WRITE PROTECTED (NO WRITE RING)#
 4283  024566  020103  042524  052123
 4284  024574  030440  035064  051440
 4285  024602  040524  052524  020123
 4286  024610  052101  041040  052117
 4287  024616  047454  026516  044514
 4288  024624  042516  053454  044522
 4289  024632  042524  050040  047522
 4290  024640  042524  052103  042105
 4291  024646  024040  047516  053440
 4292  024654  044522  042524  051040
 4293  024662  047111  024507     043
 4294  024667     045  046045  043517  MSLT15: .ASCII  /%%LOGIC TEST 15: STATUS AT BOT,OFF-LINE,WRITE PROTECTED#/
 4295  024674  041511  052040  051505
 4296  024702  020124  032461  020072
 4297  024710  052123  052101  051525
 4298  024716  040440  020124  047502
 4299  024724  026124  043117  026506
 4300  024732  044514  042516  053454
 4301  024740  044522  042524  050040
 4302  024746  047522  042524  052103
 4303  024754  042105     043
```

```
 4304  024757     045  046045  043517  MSLT16: .ASCII  /%%LOGIC TEST 16: STATUS AT EOT,ON-LINE,WRITE PROTECTED#/
 4305  024764  041511  052040  051505
 4306  024772  020124  033061  020072
 4307  025000  052123  052101  051525
 4308  025006  040440  020124  047505
 4309  025014  026124  047117  046055
 4310  025022  047111  026105  051127
 4311  025030  052111  020105  051120
 4312  025036  052117  041505  042524
 4313  025044  021504
 4314  025046  022445  047514  044507  MSLT17: .ASCII  /%%LOGIC TEST 17: STATUS AT ON-LINE,WRITE ENABLED#/
 4315  025054  020103  042524  052123
 4316  025062  030440  035067  051440
 4317  025070  040524  052524  020123
 4318  025076  052101  047440  026516
 4319  025104  044514  042516  053454
 4320  025112  044522  042524  042440
 4321  025120  040516  046102  042105
 4322  025126     043
 4323  025127     045  046045  043517  MSLT20: .ASCII  /%%LOGIC TEST 20: ILLEGAL FUNCTION TEST (M8909)#/
 4324  025134  041511  052040  051505
 4325  025142  020124  030062  020072
 4326  025150  046111  042514  040507
 4327  025156  020114  052506  041516
 4328  025164  044524  047117  052040
 4329  025172  051505  020124  046450
 4330  025200  034470  034460  021451
 4331  025206  022445  047514  044507  MSLT21: .ASCII  /%%LOGIC TEST 21: RMR(M8909)#/
 4332  025214  020103  042524  052123
 4333  025222  031040  035061  051040
 4334  025230  051115  046450  034470
 4335  025236  034460  021451
 4336  025242  022445  047514  044507  MSLT22: .ASCII  /%%LOGIC TEST 22: CPAR(M8909)#/
 4337  025250  020103  042524  052123
 4338  025256  031040  035062  041440
 4339  025264  040520  024122  034115
 4340  025272  030071  024471     043
 4341  025277     045  046045  043517  MSLT23: .ASCII  /%%LOGIC TEST 23: FMT(M8905-YB M8906)#/
 4342  025304  041511  052040  051505
 4343  025312  020124  031462  020072
 4344  025320  046506  024124  034115
 4345  025326  030071  026465  041131
 4346  025334  046440  034470  033060
 4347  025342  021451
 4348  025344  022445  047514  044507  MSLT24: .ASCII  /%%LOGIC TEST 24: DPAR(M8906 RH)#/
 4349  025352  020103  042524  052123
 4350  025360  031040  035064  042040
 4351  025366  040520  024122  034115
 4352  025374  030071  020066  044122
 4353  025402  021451
 4354  025404  022445  047514  044507  MSLT25: .ASCII  /%%LOGIC TEST 25: NEF(M8909)#/
 4355  025412  020103  042524  052123
 4356  025420  031040  035065  047040
 4357  025426  043105  046450  034470
 4358  025434  034460  021451
 4359  025440  022445  047514  044507  MSLT26: .ASCII  /%%LOGIC TEST 26: FCE(M8909)#/
```

```
 4360  025446  020103  042524  052123
 4361  025454  031040  035066  043040
 4362  025462  042503  046450  034470
 4363  025470  034460  021451
 4364  025474  022445  047514  044507  MSLT27: .ASCII  /%%LOGIC TEST 27: ILR(M8909)#/
 4365  025502  020103  042524  052123
 4366  025510  031040  035067  044440
 4367  025516  051114  046450  034470
 4368  025524  034460  021451
 4369  025530  022445  047514  044507  MSLT30: .ASCII  /%%LOGIC TEST 30:DTE(M8906 RH)#/
 4370  025536  020103  042524  052123
 4371  025544  031440  035060  052104
 4372  025552  024105  034115  030071
 4373  025560  020066  044122  021451
 4374  025566  022445  047514  044507  MSLT31: .ASCII  /%%LOGIC TEST 31: OPI(M8933)#/
 4375  025574  020103  042524  052123
 4376  025602  031440  035061  047440
 4377  025610  044520  046450  034470
 4378  025616  031463  021451
 4379  025622  022445  047514  044507  MSLT32: .ASCII  /%%LOGIC TEST 32: UNS(M8909)#/
 4380  025630  020103  042524  052123
 4381  025636  031440  035062  052440
 4382  025644  051516  046450  034470
 4383  025652  034460  021451
 4384  025656  022445  047514  044507  MSLT33: .ASCII  /%%LOGIC TEST 33: PIP(M8909)#/
 4385  025664  020103  042524  052123
 4386  025672  031440  035063  050040
 4387  025700  050111  046450  034470
 4388  025706  034460  021451
 4389  025712  022445  047514  044507  MSLT34: .ASCII  /%%LOGIC TEST 34: PES(M8931)#/
 4390  025720  020103  042524  052123
 4391  025726  031440  035064  050040
 4392  025734  051505  046450  034470
 4393  025742  030463  021451
 4394  025746  022445  047514  044507  MSLT35: .ASCII  /%%LOGIC TEST 35: SAC(M8933 M8905-YB)#/
 4395  025754  020103  042524  052123
 4396  025762  031440  035065  051440
 4397  025770  041501  046450  034470
 4398  025776  031463  046440  034470
 4399  026004  032460  054455  024502
 4400  026012     043
 4401  026013     045  046045  043517  MSLT36: .ASCII  /%%LOGIC TEST 36: FCS(M8933 M8905-YB)#/
 4402  026020  041511  052040  051505
 4403  026026  020124  033063  020072
 4404  026034  041506  024123  034115
 4405  026042  031471  020063  034115
 4406  026050  030071  026465  041131
 4407  026056  021451
 4408  026060  022445  047514  044507  MSLT37: .ASCII  /%%LOGIC TEST 37: ACCL(M8933 M8905-YB)#/
 4409  026066  020103  042524  052123
 4410  026074  031440  035067  040440
 4411  026102  041503  024114  034115
 4412  026110  031471  020063  034115
 4413  026116  030071  026465  041131
 4414  026124  021451
 4415  026126  022445  047514  044507  MSLT40: .ASCII  /%%LOGIC TEST 40: PE TAPE MARK(M8932)#/
```

```
 4416  026134  020103  042524  052123
 4417  026142  032040  035060  050040
 4418  026150  020105  040524  042520
 4419  026156  046440  051101  024113
 4420  026164  034115  031471  024462
 4421  026172     043
 4422  026173     045  046045  043517  MSLT41: .ASCII  /%%LOGIC TEST 41: NRZ TAPE MARK (M8934)#/
 4423  026200  041511  052040  051505
 4424  026206  020124  030464  020072
 4425  026214  051116  020132  040524
 4426  026222  042520  046440  051101
 4427  026230  020113  046450  034470
 4428  026236  032063  021451
 4429  026242  022445  047514  044507  MSLT42: .ASCII  /%%LOGIC TEST 42: CRC(M8934)#/
 4430  026250  020103  042524  052123
 4431  026256  032040  035062  041440
 4432  026264  041522  046450  034470
 4433  026272  032063  021451
 4434  026276  022445  047514  044507  MSLT43: .ASCII  /%%LOGIC TEST 43: LRC(M8934)#/
 4435  026304  020103  042524  052123
 4436  026312  032040  035063  046040
 4437  026320  041522  046450  034470
 4438  026326  032063  021451
 4439  026332  022445  047514  044507  MSLT44: .ASCII  /%%LOGIC TEST 44: CORRECTABLE DATA (M8932 M8901)#/
 4440  026340  020103  042524  052123
 4441  026346  032040  035064  041440
 4442  026354  051117  042522  052103
 4443  026362  041101  042514  042040
 4444  026370  052101  020101  046450
 4445  026376  034470  031063  046440
 4446  026404  034470  030460  021451
 4447  026412  022445  047514  044507  MSLT45: .ASCII  /%%LOGIC TEST 45: INCORRECTABLE DATA (M8932 M8934)#/
 4448  026420  020103  042524  052123
 4449  026426  032040  035065  044440
 4450  026434  041516  051117  042522
 4451  026442  052103  041101  042514
 4452  026450  042040  052101  020101
 4453  026456  046450  034470  031063
 4454  026464  046440  034470  032063
 4455  026472  021451
 4456  026474  022445  047514  044507  MSLT46: .ASCII  /%%LOGIC TEST 46: PEF(M8932)#/
 4457  026502  020103  042524  052123
 4458  026510  032040  035066  050040
 4459  026516  043105  046450  034470
 4460  026524  031063  021451
 4461  026530  022445  047514  044507  MSLT47: .ASCII  /%%LOGIC TEST 47: FC OVERFLOW (M8905-YB)#/
 4462  026536  020103  042524  052123
 4463  026544  032040  035067  043040
 4464  026552  020103  053117  051105
 4465  026560  046106  053517  024040
 4466  026566  034115  030071  026465
 4467  026574  041131  021451
 4468  026600  022445  047514  044507  MSLT50: .ASCII  /%%LOGIC TEST 50: NEF WHEN WRITE PE ON NRZ SLAVE#/
 4469  026606  020103  042524  052123
 4470  026614  032440  035060  047040
 4471  026622  043105  053440  042510
```

```
 4472  026630  020116  051127  052111
 4473  026636  020105  042520  047440
 4474  026644  020116  051116  020132
 4475  026652  046123  053101  021505
 4476  026660  022445  047514  044507  MSL151: .ASCII  /%%LOGIC TEST 51: NEF WHEN WRITE NRZ ON PE SLAVE#/
 4477  026666  020103  042524  052123
 4478  026674  032440  035061  047040
 4479  026702  043105  053440  042510
 4480  026710  020116  051127  052111
 4481  026716  020105  051116  020132
 4482  026724  047117  050040  020105
 4483  026732  046123  053101  021505
```

```
4484
4485                                                    ;MANUAL INSTRUCTION**********
4486
4487  026740  052045  050131  020105  MMSG0:  .ASCII  /%TYPE CR WHEN READY;#/
4488  026746  051103  053440  042510
4489  026754  020116  042522  042101
4490  026762  035531     043
4491  026765     045  046445  052517  MMSG1:  .ASCII  /%%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE:#/
4492  026772  052116  052040  050101
4493  027000  020105  044527  044124
4494  027006  047040  020117  051127
4495  027014  052111  020105  044522
4496  027022  043516  020054  047514
4497  027030  042101  052040  020117
4498  027036  047502  026124  051440
4499  027044  052105  052040  020117
4500  027052  047117  046040  047111
4501  027060  035105     043
4502  027063     045  042523  020124  MMSG2:  .ASCII  /%SET TO OFFLINE:#/
4503  027070  047524  047440  043106
4504  027076  044514  042516  021472
4505  027104  046445  053117  020105  MMSG3:  .ASCII  /%MOVE FORWARD TO EOT, ONLINE:#/
4506  027112  047506  053522  051101
4507  027120  020104  047524  042440
4508  027126  052117  020054  047117
4509  027134  044514  042516  021472
4510  027142  047445  043106  046040  MMSG4:  .ASCII  /%OFF LINE REVERSE PAST EOT, INSERT WRITE RING, ON LINE#/
4511  027150  047111  020105  042522
4512  027156  042526  051522  020105
4513  027164  040520  052123  042440
4514  027172  052117  020054  047111
4515  027200  042523  052122  053440
4516  027206  044522  042524  051040
4517  027214  047111  026107  047440
4518  027222  020116  044514  042516
4519  027230     043
4520  027231     045  046445  053117  MMSG5:  .ASCII  /%%MOVE TAPE TO BOT; ON LINE#/
4521  027236  020105  040524  042520
4522  027244  052040  020117  047502
4523  027252  035524  047440  020116
4524  027260  044514  042516     043
```

```
4525
4526                                                ;TAG MESSAGE
4527
4528  027265     045  053523  020122  $MSWR:  .ASCII  /%SWR = #/
4529  027272  020075     043
4530  027275     040  042516  020127  $MNEW:  .ASCII  / NEW = #/
4531  027302  020075     043
4532  027305     045  046123  020101  TMS1:   .ASCII  /%SLA #/
4533  027312     043
4534  027313     045  047502  020124  TMS2:   .ASCII  /%BOT #/
4535  027320     043
4536  027321     045  046524  021440  TMS3:   .ASCII  /%TM #/
4537  027326  044445  041104  021440  TMS4:   .ASCII  /%IDB #/
4538  027334  051445  053504  020116  TMS5:   .ASCII  /%SDWN #/
4539  027342     043
4540  027343     045  042520  020123  TMS6:   .ASCII  /%PES #/
4541  027350     043
4542  027351     045  051523  020103  TMS7:   .ASCII  /%SSC #/
4543  027356     043
4544  027357     045  051104  020131  TMS8:   .ASCII  /%DRY #/
4545  027364     043
4546  027365     045  050104  020122  TMS9:   .ASCII  /%DPR #/
4547  027372     043
4548  027373     045  052116  020114  TMS10:  .ASCII  /%NTL #/
4549  027400     043
4550  027401     045  047505  020124  TMS11:  .ASCII  /%ECT #/
4551  027406     043
4552  027407     045  051127  020114  TMS12:  .ASCII  /%WRL #/
4553  027414     043
4554  027415     045  047515  020114  TMS13:  .ASCII  /%MOL #/
4555  027422     043
4556  027423     045  044520  020120  TMS14:  .ASCII  /%PIP #/
4557  027430     043
4558  027431     045  051105  020122  TMS15:  .ASCII  /%ERR #/
4559  027436     043
4560  027437     045  052101  020101  TMS16:  .ASCII  /%ATA #/
4561  027444     043
4562  027445     045  046111  020106  TMS17:  .ASCII  /%ILF #/
4563  027452     043
4564  027453     045  046111  020122  TMS18:  .ASCII  /%ILR #/
4565  027460     043
4566  027461       5  046522  020122  TMS19:  .ASCII  /%RMR #/
4567  027466     043
4568  027467     045  050103  051101  TMS20:  .ASCII  /%CPAR #/
4569  027474  021440
4570  027476  043045  052115  021440  TMS21:  .ASCII  /%FMT #/
4571  027504  042045  040520  020122  TMS22:  .ASCII  /%DPAR #/
4572  027512     043
4573  027513     045  047111  020103  TMS23:  .ASCII  /%INC #/
4574  027520     043
4575  027521     045  050126  020105  TMS24:  .ASCII  /%VPE #/
4576  027526     043
4577  027527     045  042520  020106  TMS25:  .ASCII  /%PEF #/
4578  027534     043
4579  027535     045  051114  020103  TMS26:  .ASCII  /%LRC #/
4580  027542     043
```

```
 4581  027543     045  051516  020107  TMS27:  .ASCII  /%NSG #/
 4582  027550     043
 4583  027551     045  041506  020105  TMS28:  .ASCII  /%FCE #/
 4584  027556     043
 4585  027557     045  051503  021440  TMS29:  .ASCII  /%CS #/
 4586  027564  044445  046524  021440  TMS30:  .ASCII  /%ITM #/
 4587  027572  047045  043105  021440  TMS31:  .ASCII  /%NEF #/
 4588  027600  042045  042524  021440  TMS32:  .ASCII  /%DTE #/
 4589  027606  047445  044520  021440  TMS33:  .ASCII  /%OPI #/
 4590  027614  053445  044522  042524  TMS33A: .ASCII  /%WRITE OPI #/
 4591  027622  047440  044520  021440
 4592  027630  051045  040505  020104  TMS33B: .ASCII  /%READ OPI #/
 4593  027636  050117  020111     043
 4594  027643     040  041517  052503  TMS33C: .ASCII  / OCCURED TO SOON%#/
 4595  027650  042522  020104  047524
 4596  027656  051440  047517  022516
 4597  027664     043
 4598  027665     040  041517  052503  TMS33D: .ASCII  / OCCURRED TO LATE%#/
 4599  027672  051122  042105  052040
 4600  027700  020117  040514  042524
 4601  027706  021445
 4602  027710  043040  044501  042514  TMS33E: .ASCII  / FAILED TO SET%#/
 4603  027716  020104  047524  051440
 4604  027724  052105  021445
 4605  027730  052445  051516  021440  TMS34:  .ASCII  /%UNS #/
 4606  027736  041445  051117  020122  TMS35:  .ASCII  /%CORR #/
 4607  027744     043
 4608  027745     045  051103  020103  TMS36:  .ASCII  /%CRC #/
 4609  027752     043
 4610  027753     045  040523  020103  TMS37:  .ASCII  /%SAC #/
 4611  027760     043
 4612  027761     045  041506  020123  TMS38:  .ASCII  /%FCS #/
 4613  027766     043
 4614  027767     045  041501  046103  TMS39:  .ASCII  /%ACCL #/
 4615  027774  021440
 4616
 4617                                          .EVEN
 4618                                          ;WRITE BUFFER
 4619
 4620  027776  000100                  WDATA:
 4621  027776  177777                          -1
 4622  030000  177777                          -1
 4623  030002  177777                          -1
 4624  030004  177777                          -1
 4625  030006  177777                          -1
 4626  030010  177777                          -1
 4627  030012  177777                          -1
 4628  030014  177777                          -1
 4629  030016  177777                          -1
 4630  030020  177777                          -1
 4631  030022  177777                          -1
 4632  030024  177777                          -1
 4633  030026  177777                          -1
 4634  030030  177777                          -1
 4635  030032  177777                          -1
 4636  030034  177777                          -1
```

```
 4637  030036  177777                    -1
 4638  030040  177777                    -1
 4639  030042  177777                    -1
 4640  030044  177777                    -1
 4641  030046  177777                    -1
 4642  030050  177777                    -1
 4643  030052  177777                    -1
 4644  030054  177777                    -1
 4645  030056  177777                    -1
 4646  030060  177777                    -1
 4647  030062  177777                    -1
 4648  030064  177777                    -1
 4649  030066  177777                    -1
 4650  030070  177777                    -1
 4651  030072  177777                    -1
 4652  030074  177777                    -1
 4653  030076  177777                    -1
 4654  030100  177777                    -1
 4655  030102  177777                    -1
 4656  030104  177777                    -1
 4657  030106  177777                    -1
 4658  030110  177777                    -1
 4659  030112  177777                    -1
 4660  030114  177777                    -1
 4661  030116  177777                    -1
 4662  030120  177777                    -1
 4663  030122  177777                    -1
 4664  030124  177777                    -1
 4665  030126  177777                    -1
 4666  030130  177777                    -1
 4667  030132  177777                    -1
 4668  030134  177777                    -1
 4669  030136  177777                    -1
 4670  030140  177777                    -1
 4671  030142  177777                    -1
 4672  030144  177777                    -1
 4673  030146  177777                    -1
 4674  030150  177777                    -1
 4675  030152  177777                    -1
 4676  030154  177777                    -1
 4677  030156  177777                    -1
 4678  030160  177777                    -1
 4679  030162  177777                    -1
 4680  030164  177777                    -1
 4681  030166  177777                    -1
 4682  030170  177777                    -1
 4683  030172  177777                    -1
 4684  030174  177777                    -1
 4685
 4686
 4687                                    ;READ BUFFER
 4688
 4689  030176  000100          RDATA:
 4690  030176  000000                    0
 4691  030200  000000                    0
 4692  030202  000000                    0
```

4693	030204	000000	0
4694	030206	000000	0
4695	030210	000000	0
4696	030212	000000	0
4697	030214	000000	0
4698	030216	000000	0
4699	030220	000000	0
4700	030222	000000	0
4701	030224	000000	0
4702	030226	000000	0
4703	030230	000000	0
4704	030232	000000	0
4705	030234	000000	0
4706	030236	000000	0
4707	030240	000000	0
4708	030242	000000	0
4709	030244	000000	0
4710	030246	000000	0
4711	030250	000000	0
4712	030252	000000	0
4713	030254	000000	0
4714	030256	000000	0
4715	030260	000000	0
4716	030262	000000	0
4717	030264	000000	0
4718	030266	000000	0
4719	030270	000000	0
4720	030272	000000	0
4721	030274	000000	0
4722	030276	000000	0
4723	030300	000000	0
4724	030302	000000	0
4725	030304	000000	0
4726	030306	000000	0
4727	030310	000000	0
4728	030312	000000	0
4729	030314	000000	0
4730	030316	000000	0
4731	030320	000000	0
4732	030322	000000	0
4733	030324	000000	0
4734	030326	000000	0
4735	030330	000000	0
4736	030332	000000	0
4737	030334	000000	0
4738	030336	000000	0
4739	030340	000000	0
4740	030342	000000	0
4741	030344	000000	0
4742	030346	000000	0
4743	030350	000000	0
4744	030352	000000	0
4745	030354	000000	0
4746	030356	000000	0
4747	030360	000000	0
4748	030362	000000	0

```
 4749  030364  000000                          0
 4750  030366  000000                          0
 4751  030370  000000                          0
 4752  030372  000000                          0
 4753  030374  000000                          0
 4754
 4755                                          ;WRAP AROUND MESSAGES******
 4756
 4757  030376  042045  052101  020101  WMSG27: .ASCII  /%DATA PAT:#/
 4758  030404  040520  035124    043
 4759  030411     045  047505  020122  WMSG31: .ASCII  /%EOR CLEAR DID NOT CLEAR GO%#/
 4760  030416  046103  040505  020122
 4761  030424  044504  020104  047516
 4762  030432  020124  046103  040505
 4763  030440  020122  047507  021445
 4764
 4765                                          .EVEN
 4766  030446  000000                  PRF:    0
 4767  030450  000000                          0
 4768  030452  000000                          0
 4769  030454  000000                          0
 4770  030456  000000                          0
 4771  030460  000000                          0
 4772  030462  000000                          0
 4773  030464  000000                          0
 4774  030466  000000                          0
 4775  030470  000000                          0
 4776  030472  000000                          0
 4777  030474  000000                          0
 4778  030476  000000                          0
 4779  030500  000000                          0
 4780  030502  000000                          0
 4781  030504  000000                          0
 4782  030506  000000                          0
 4783  030510  000000                          0
 4784  030512  000000                          0
 4785  030514  000000                          0
 4786  030516  000000                          0
 4787  030520  000000                          0
 4788  030522  000000                          0
 4789  030524  000000                          0
 4790  030526  000000                          0
 4791  030530  000000                          0
 4792  030532  000000                          0
 4793  030534  000000                          0
 4794  030536  000000                          0
 4795  030540  000000                          0
 4796  030542  000000                          0
 4797  030544  000000                          0
 4798  030546  000000                          0
 4799  030550  000000                          0
 4800  030552  000000                          0
 4801  030554  000000                          0
 4802  030556  000000                          0
 4803  030560  000000                          0
 4804  030562  000000                          0
```

```
 4805  030564  000000                        0
 4806  030566  000000                        0
 4807  030570  000000                POST:   0
 4808  030572  000000                        0
 4809  030574  000000                        0
 4810  030576  000000                        0
 4811  030600  000000                        0
 4812  030602  000000                        0
 4813  030604  000000                        0
 4814  030606  000000                        0
 4815  030610  000000                        0
 4816  030612  000000                        0
 4817  030614  000000                        0
 4818  030616  000000                        0
 4819  030620  000000                        0
 4820  030622  000000                        0
 4821  030624  000000                        0
 4822  030626  000000                        0
 4823  030630  000000                        0
 4824  030632  000000                        0
 4825  030634  000000                        0
 4826  030636  000000                        0
 4827  030640  000000                        0
 4828  030642  000000                        0
 4829  030644  000000                        0
 4830  030646  000000                        0
 4831  030650  000000                        0
 4832  030652  000000                        0
 4833  030654  000000                        0
 4834  030656  000000                        0
 4835  030660  000000                        0
 4836  030662  000000                        0
 4837  030664  000000                        0
 4838  030666  000000                        0
 4839  030670  000000                        0
 4840  030672  000000                        0
 4841  030674  000000                        0
 4842  030676  000000                        0
 4843  030700  000000                        0
 4844  030702  000000                        0
 4845  030704  000000                        0
 4846  030706  000000                        0
 4847  030710  000000                        0
 4848  030712  000000            WBUFF:      0
 4849          031324
 4850  031324  000000            RBUFF:      0
 4851                                        .=.+4^O
 4852          000001                        .END
```

```
ADDFL   000742  1401#  3260*  3272   3284   3312*
AS      000526  1323#  3335   3370   3413
ASF     000724  1394#
ATAF    000714  1390#
ATTA    016100  3399   3402#
ATTB    016120  3403   3406#
ATTC    016140  3410#  3412
ATTD    016146  3409   3413#
ATTN    016060  3351   3397#
ATTP    016156  3401   3405   3414   3416#  3433
ATTPA   016204  3419   3422#
ATTX    016230  3417   3427#
ATTXA   016240  3428   3430#
ATTXX   016256  3431   3434#
BA      000514  1318#  2343*  2906*  2927*  2965*  3002*  3047*  3091*  3131*  3164*  3195*  3325
CC      000530  1324#  2916   3021   3023   3065
CHNFLG  001420  1552#  1557*  1568   1667   1715   1732   3611
CRCNT   000770  1412#
CS      000520  1320#  1671*  1675*  1677   1764*  1765*  1767   1792*  1793*  1883*  1884*  2026*  2027*
                2294*  2297*  2348*  2386*  3329   3531*  3532*  3554*  3555*
C1      000510  1316#  1595   1676   1766   1795   1800   1837   2028*  2029   2051   2057*  2058   2062
                2096*  2123*  2125*  2241*  2267*  2321*  2347*  2400*  2428*  2459   2490*  2500*  2550*
                2573*  2697*  2724*  2764*  2814*  2836*  2856*  2908*  2929*  2966*  3003*  3048*  3092*
                3133*  3166*  3197*  3235   3321   3352*  3566*  3574*  3579*
DATAD   000754  1406#
DATC    000750  1404#
DB      000532  1325#
DCA     015656  3354#  3357
DCAO    015672  3355   3358#
DCB     015730  3367#  3369
DCC     015736  3366   3370#
DCD     015632  3348#  3349
DCE     015746  3359   3361   3363   3371   3374#  3391
DCEA    015776  3378   3381#
DCEX    016016  3376   3385#
DCEXA   016026  3386   3388#
DCEXX   016044  3389   3392#
DERFL   000762  1409#
DOUT    020232  3809#  3825   3827
DOUTD   020310  3823#
DRVCLR  015624  2257   2285   2309   2334   2388   2414   2454   2477   2530   2584   2709   2887   2947
                2991   3035   3079   3121   3181   3212   3346#  3393
DRVN    000620  1360#  1617   1619   1669*  1672*  1673   1675   1681   1743   1744   1793   1884   2027
                3365   3408   3532   3555
DS      000522  1321#  2091   2097   2100   2151   2173   2194   2217   2725   2741   2747   2760   2838
                2858   2882   2910   2931   2968   3011   3056   3101   3331   3354   3358   3362   3398
                3402   3561   3564   3567   3569   3572   3575   3577
DT      000536  1327#  1689   1886   1898   1902   2688   2695   2952   3114
EMADDR  000616  1359#  1742*  1794*  1829*  1925*  1934*  1973*  1999*  2023*  2049*  2089*  2120*  2146*
                2168*  2189*  2212*  2234*  2262*  2291*  2314*  2338*  2395*  2419*  2462*  2483*  2506
                2536*  2625   2649   2660   2683*  2719*  2736*  2758*  2782*  2802*  2829*  2851*  2901*
                2957*  2997*  3042*  3086*  3125*  3157*  3188*  3245   3267   3379   3420   3445
EORPA   015010  2284   2333   2387   2529   3228#
EORPX   015136  3253#
EORP1   015044  3235#  3238   3240
EORP1A  015112  3244   3247#
```

```
EOKP2   015134  3236  3242  3250  3252#
ER      000524  1322# 1803  1834  2242  2270  2298  2322  2368  2381  2402  2437  2445  2449
                2467  2491  2517  2523  2551  2554  2574  2577  2698  2865  2870  2877  2914
                2935  2942  2972  2985  3015  3029  3060  3072  3105  3111  3171  3202  3333
                3360
ERADD   000666  1379# 1783* 1812* 1815* 1818* 1843* 1848* 1926* 1959* 1963* 1988* 2012* 2039*
                2066* 2072* 2078* 2103* 2107* 2111* 2129* 2155* 2177* 2199* 2222* 2245* 2273*
                2301* 2325* 2375* 2406* 2445* 2470* 2482* 2538* 2563* 2627  2651  2662  2701*
                2727* 2735* 2757* 2783* 2803* 2842* 2862* 2867* 2872* 2937* 2974* 2983* 3017*
                3026* 3062* 3070* 3107* 3148* 3259* 3270  3448
ERRF    000722  1393#
EXFL    000712  1389# 1780* 1782* 1811* 1814* 1817* 1922* 1924* 2246* 2274* 2302* 2326* 2376*
                2407* 2446* 2469* 2493* 2519* 2702* 2728* 2743* 2750* 2767* 2775* 2788* 2795*
                2808* 2821* 2843* 2863* 2868* 2873* 2938* 2975* 3018* 3063* 3108* 3149* 3174*
                3205* 3258* 3276  3279  3313*
FC      000516  1319# 1833* 1836  1845  2002* 2003  2055* 2094* 2268* 2295* 2344* 2398* 2426*
                2489* 2498* 2546* 2723* 2791* 2854* 2905* 2926* 2964* 3001* 3005  3046* 3050
                3090* 3093  3129* 3144  3163* 3194* 3327
FUN     000746  1403#
GTSWR   020436  1657  3628  3859#
HDRFL   000614  1358# 1695* 3243  3265  3269* 3377  3418  3426* 3443  3447*
HERE    002626  1725  1731#
ILFT    000544  1333# 2237
INIT    016740  2903  2924  2958  2999  3044  3088  3553#
INIT1   016662  1832  1935  1976  2001  2050  2061  2090  2099  2121  2149  2238  2264  2293
                2316  3530#
INIT2   016672  2171  2192  2215  3532#
INIT3   016710  2341  2396  2420  2465  2485  2495  2540  2564  2685  2721  2738  2759  2770
                2784  2790  2804  2811  2852  3127  3159  3542#
INIT4   016724  2746  2830  3190  3547#
INIT5   016744  3544  3549  3554#
INMT    016604  2145  2167  2188  2211  3509#
INST    017134  1752  1871  2148  2170  2191  2214  3587#
ITAMT   000602  1351# 2340* 2390* 2539* 2585* 3501
ITCNT   000676  1383# 3303  3464  3496* 3500* 3501
ITER    016534  1824  1857  1967  1992  2017  2043  2083  2114  2132  2157  2179  2201  2224
                2256  2286  2308  2332  2389  2413  2453  2476  2528  2583  2708  2730  2752
                2777  2797  2823  2845  2886  2946  2990  3034  3078  3120  3151  3180  3211
                3494#
ITRLP   000710  1388# 1693* 1694* 3504
JUMPER  001016  1423# 1607  1612
LTADD   000736  1399# 1691* 1692* 1693  1696  1712* 1713*
LTGA    015202  3265# 3311
LTGA1   015220  3266  3269#
LTGA2   015252  3273  3276#
LTGB    015300  3280  3282#
LTGC    015304  3277  3283#
LTGCO   015372  3287  3298#
LTGD    015376  3285  3299#
LTGER   015164  1785  1820  1928  2471  3261#
LTGER0  015156  2247  2275  2303  2327  2377  2408  2447  2494  2520  2703  2729  2745  2751
                2769  2776  2789  2796  2810  2822  2844  2864  2869  2874  2939  2976  3019
                3064  3109  3175  3206  3260#
LTGER1  016272  1846  1850  1961  1965  1990  2014  2041  2070  2076  2082  2156  2178  2200
                2223  2984  3027  3071  3150  3439#
LTGER2  016264  2105  2109  2113  2131  3438#
```

```
LTGER3  015144          2251  2279  2307  2331  2385  2412  2452  2527  2557  2580  2707  2880  2920
                        2945  2989  3033  3077  3119  3179  3210  3258#
LTGX    015426          3264  3300  3305#
LTGXA   015436          3306  3308#
LTGXX   015454          3309  3312#
LTG1A   016310          3443# 3474
LTG1B   016326          3444  3447#
LTG1C   016436          3461  3466#
LTG1T   016406          3451  3460#
LTG1X   016442          3442  3467#
LTG1XX  016472          3472  3475#
LTG1X1  016454          3469  3471#
LT1     002666          1450  1451  1742#
LT1A    003012          1750  1764# 1779  1784
LT1B    003060          1768  1774#
LT1C    003066          1772  1776# 1786
LT1ER   003076          1773  1780#
LT1ER1  003106          1775  1782#
LT1ER2  003114          1781  1783#
LT1G    002742          1753# 1777
LT1G0   002732          1751#
LT1X    003136          1748  1761  1770  1787#
LT10    004552          1464  2022#
LT10A   004570          2026# 2038  2040
LT10A1  004566          2025# 2036
LT10B   004624          2033# 2042
LT10E1  004644          2032  2039#
LT10IT  004554          1465  2023#
LT10X   004666          2034  2043#
LT11    004676          1466  2048#
LT11B   004724          2054# 2071  2075
LT11C   004764          2061# 2077  2081
LT11E1  005004          2053  2066#
LT11E2  005034          2060  2072#
LT11E3  005064          2065  2078#
LT11IT  004700          1467  2049# 2069
LT11X   005112          2064  2083#
LT12    005122          1468  2088#
LT12B   005146          2093# 2106  2108
LT12C   005204          2099# 2110  2112
LT12E1  005222          2092  2103#
LT12E2  005244          2098  2107#
LT12E3  005266          2102  2111#
LT12IT  005124          1469  2089# 2104
LT12X   005306          2101  2114#
LT13    005316          1470  2119#
LT13A   005356          2126# 2127
LT13E1  005362          2128#
LT13IT  005326          1471  2121# 2130
LT13X   005410          2122  2132#
LT14    005420          1472  2141#
LT14A   005442          2142  2146#
LT14IT  005460          1473  2149# 2154
LT14X   005520          2153  2157#
LT14XX  005524          2144  2158#
LT15    005530          1474  2163#
```

```
LT15A   005552          2164    2168#
LT15IT  005570          1475    2171#   2176
LT15X   005630          2175    2179#
LT15XX  005634          2166    2180#
LT16    005640          1476    2184#
LT16A   005662          2185    2189#
LT16IT  005700          1477    2192#   2198
LT16X   005744          2197    2201#
LT16XX  005750          2187    2202#
LT17    005754          1478    2207#
LT17A   005776          2208    2212#
LT17IT  006014          1479    2215#   2221
LT17X   006060          2220    2224#
LT17XX  006064          2210    2225#
LT2     003142          1452    1453    1791#
LT2A    003202          1798#   1808    1823
LT2B    003222          1801    1803#
LT2C    003236          1804    1806#
LT2ERG  003316          1813    1816    1819#
LT2ER1  003246          1802    1811#
LT2ER2  003264          1805    1814#
LT2ER3  003302          1817#
LT2IT   003144          1792#
LT2LP   003332          1819    1822#
LT2X    003336          1809    1824#
LT20    006070          1480    2234#
LT20A   006116          2235    2238#   2255
LT20B   006174          2244    2249#
LT20C   006204          2248    2250    2252#
LT20IT  006104          1481    2236#
LT20X   006216          2253    2256#
LT21    006232          1482    2262#
LT21A   006336          2272    2277#
LT21B   006346          2276    2278    2281#
LT21IT  006246          1483    2263    2264#
LT21XA  006352          2282#   2283
LT22    006376          1484    2291#
LT22A   006476          2300    2305#
LT22IT  006412          1485    2292    2293#
LT22X   006506          2304    2306    2308#
LT23    006522          1486    2314#
LT23A   006622          2324    2329#
LT23IT  006536          1487    2315    2316#
LT23X   006632          2328    2330    2332#
LT24    006652          1488    2338#
LT24B   007000          2357#   2363
LT24B0  007022          2360    2362#
LT24C   007034          2366#   2367
LT24D   007116          2369    2380#
LT24IT  006666          1489    2339    2340#
LT24X   007142          2379    2384    2386#
LT25    007176          1490    2395#
LT25A   007304          2404    2410#
LT25IT  007204          1491    2396#   2405
LT25X   007314          2409    2411    2413#
LT26    007330          1492    2419#
```

```
LT26IT  007336  1493  2420#  2444
LT26X   007530  2448  2451   2453#
LT27    007544  1494  2459#
LT27A   007600  2461  2465#  2475
LT27B   007636  2468  2472#
LT27IT  007570  1495  2463#
LT27X   007646  2473  2476#
LT27XX  007656  2460  2478#
LT3     003346  1454  1828#
LT3A    003366  1832#  1842  1844  1856
LT3B    003406  1836#  1847  1849
LT3C    003422  1839#  1851
LT3ER1  003432  1835  1843#
LT3ER2  003460  1838  1848#
LT3IT   003350  1455  1829#
LT3X    003502  1840  1852#
LT3XX   003520  1853  1857#
LT30    007662  1496  2482#
LT30A   007764  2492  2495#
LT30B   010026  2502#  2505
LT30C   010066  2503  2512#
LT30D   010104  2515#  2516
LT30E   010134  2518  2522#
LT30IT  007704  1497  2484  2485#
LT30X   010160  2511  2521  2526  2528#
LT31    010200  1498  2536#
LT31A   010356  2562  2564#
LT31IT  010206  1499  2537#
LT31X   010462  2558  2560  2581  2583#  2631
LT32    011030  1500  2683#
LT32IT  011044  1501  2684  2685#
LT32X   011172  2704  2706  2708#
LT32XX  011202  2696  2710#
LT33    011206  1502  2719#
LT33IT  011222  1503  2720  2721#
LT33X   011300  2726  2730#
LT34    011310  1504  2735#
LT34A   011350  2741#
LT34A1  011330  2738#  2744
LT34B   011400  2742  2746#
LT34C   011404  2747#  2749
LT34IT  011324  1505  2737#
LT34X   011434  2748  2752#
LT34XX  011440  2753#
LT35    011444  1506  2757#
LT35A   011546  2766  2770#  2774
LT35IT  011460  1507  2759#  2768
LT35X   011606  2773  2777#
LT36    011616  1508  2782#
LT36IT  011632  1509  2784#  2787
LT36X   011726  2793  2797#
LT37    011736  1510  2802#
LT37A   012012  2807  2811#  2820
LT37B   012044  2816#  2819
LT37IT  011752  1511  2804#  2809
LT37X   012076  2817  2823#
```

```
LT4     003530            1456    1457    1862#
LT4A    003646            1869    1883#   1921    1927
LT4B    003710            1888    1892#
LT4C    003716            1894#
LT4D    004050            1890    1918#   1929
LT4ERG  004076            1923    1925#
LT4ER1  004060            1891    1922#
LT4ER2  004070            1893    1924#
LT4G    003576            1872#   1919
LT4G0   003566            1870#
LT4X    004126            1867    1880    1917    1930#
LT40    012106            1512    2828#
LT40IT  012122            1513    2828    2830#
LT40X   012216            2839    2845#
LT40XX  012222            2846#
LT41    012226            1514    2850#
LT41IT  012242            1515    2850    2852#
LT41X   012436            2875    2879    2881#
LT42    012470            1516    2896#
LT42A   012572            2910#   2913
LT42B   012606            2911    2914#
LT42B1  012634            2915    2920#
LT42B2  012654            2919    2924#
LT42C   012720            2931#   2934
LT42D   012734            2932    2935#
LT42E   012764            2936    2941#
LT42IT  012526            1517    2902    2903#
LT42X   013004            2940    2944    2946#
LT43    013020            1518    2952#
LT43C   013130            2968#   2971
LT43D   013144            2969    2972#
LT43E   013176            2973    2978#
LT43F   013230            2982    2985#
LT43IT  013056            1519    2956    2958#
LT43X   013250            2977    2988    2990#
LT43XX  013260            2953    2992#
LT44    013264            1520    2995#
LT44A   013350            3005#   3008
LT44A1  013362            3006    3009#
LT44B   013372            3011#   3014
LT44C   013406            3012    3015#
LT44D   013434            3016    3020#
LT44E   013436            3021#
LT44F   013476            3022    3029#
LT44IT  013312            1521    2998    2999#
LT44X   013516            3028    3032    3034#
LT44XX  013526            3036#
LT45    013532            1522    3040#
LT45A   013640            3056#   3059
LT45B   013654            3057    3060#
LT45D   013704            3061    3065#
LT45E   013616            3050#   3053
LT45E1  013630            3051    3054#
LT45F   013736            3069    3072#
LT45IT  013560            1523    3043    3044#
LT45X   013762            3076    3078#
```

```
LT45XX  013772            3080#
LT46    013776            1524    3084#
LT46A   014060            3093#   3094
LT46B   014116            3101#   3104
LT46C   014132            3102    3105#
LT46D   014164            3106    3111#
LT46IT  014024            1525    3087    3088#
LT46X   014224            3110    3118    3120#
LT46XX  014234            3122#
LT47    014240            1526    3125#
LT47A   014330            3135#   3138
LT47B   014342            3136    3139#
LT47C   014346            3140#   3143
LT47IT  014254            1527    3126    3127#
LT47X   014420            3147    3151#
LT5     004132            1458    1934#
LT5A    004152            1938#   1946    1960
LT5B    004172            1943#   1962
LT5C    004202            1944    1947#
LT5D    004212            1949#   1958    1964
LT5E    004234            1953    1955#   1966
LT5ER1  004246            1942    1959#
LT5ER2  004270            1954    1963#
LT5IT   004140            1459    1935#
LT5X    004312            1956    1967#
LT50    014430            1528    3157#
LT50IT  014444            1529    3158    3159#
LT50X   014570            3176    3178    3180#
LT51    014604            1530    3188#
LT51IT  014620            1531    3189    3190#
LT51X   014744            3207    3209    3211#
LT6     004322            1460    1972#
LT6A    004340            1976#   1987
LT6A1   004336            1975#   1985
LT6B    004344            1977#   1989
LT6D    004364            1982#   1991
LT6ER1  004404            1981    1988#
LT6IT   004324            1461    1973#
LT6X    004426            1983    1992#
LT7     004436            1462    1997#
LT7A    004454            2001#   2011    2013
LT7B    004474            2006#   2015
LT7C    004444            1999#   2009
LT7ER1  004514            2005    2012#
LT7IT   004440            1463    1998#
LT7X    004536            2007    2016#
MMSG0   026740            3589    4487#
MMSG1   026765            2147    4491#
MMSG2   027063            2169    4502#
MMSG3   027104            2190    4505#
MMSG4   027142            2213    4510#
MMSG5   027231            4520#
MR      000534            1326#   1938*   1939    1949*   1950    2054*   2093*   2266*   2320*   2346*   2357*   2361    2364
                          2399*   2427*   2433*   2434*   2488*   2499*   2512*   2513*   2545*   2569*   2595    2597    2602
                          2610    2621    2722*   2763*   2913*   2835*   2855*   2907*   2928*   2962*   2978    3009*   3054*
                          3095    3097    3099*   3132*   3140*   3141*   3165*   3196*   3228    3231*   3232*   3337
```

MSG1	020600	1566	1571*	3898#		
MSG10	021522	1848	3984#			
MSG11	021546	1843	3988#			
MSG12	021573	3290	3452	3992#		
MSG13	021602	3294	3456	3994#		
MSG14	021611	1959	3996#			
MSG15	021626	1963	3999#			
MSG16	021644	3301	3462	4002#		
MSG18	021654	1988	4004#			
MSG19	021673	2012	3148	4007#		
MSG2	021003	1753	3921#			
MSG2A	021044	1751	3927#			
MSC20	021712	2039	4010#			
MSG21	021744	2066	4015#			
MSG22	022002	2072	4020#			
MSG23	022023	2078	4023#			
MSG24	022056	2103	4028#			
MSG25	022104	2107	4032#			
MSG25A	022133	2111	4036#			
MSG26	022160	2129	4040#			
MSG27	022207	2155	2177	2199	2222	4044#
MSG3	021173	1783	3943#			
MSG30	022223	1912	4047#			
MSG31	022231	3404	4049#			
MSG32	022247	3400	4052#			
MSG33	022265	3407	4055#			
MSG34	022306	4058#				
MSG35	022323	4061#				
MSG36	022341	4064#				
MSG37	022357	4067#				
MSG4	021215	1812	1818	3947#		
MSG40	022375	3705	4070#			
MSG41	022401	1717	4071#			
MSG42	022420	3026	4074#			
MSG43	022435	3510	4077#			
MSG44	022541	1572	4089#			
MSG45	022563	1581	4093#			
MSG46	022605	3319	4097#			
MSG47	022711	3381	4109#			
MSG5	021233	1815	3950#			
MSG50	022743	2508	4114#			
MSG51	022762	3259	4117#			
MSG53	023011	2983	4121#			
MSG54	023023	3070	4123#			
MSG55	023034	2921	4125#			
MSG56	023131	1633	4136#			
MSG57	023156	1615	4140#			
MSG58	023335	1624	4160#			
MSG59	023360	3665	4164#			
MSG6	021253	3278	3953#			
MSG60	023362	3674	4165#			
MSG62	023364	1563	4166#			
MSG63	023432	3622	4173#			
MSG64	023460	1894	4177#			
MSG65	023476	1897	4180#			
MSG66	023503	1900	4181#			

```
MSG67   023510            1643    4182#
MSG68   023546            1909    4187#
MSG69   023620            1905    4194#
MSG7    021272            3281    3956#
MSG8    021310            1872    3959#
MSG8A   021351            1870    3965#
MSG9    021500            1926    3981#
MSLT1   023630            1742    4198#
MSLT10  024317            2023    4253#
MSLT11  024375            2049    4261#
MSLT12  024442            2089    4268#
MSLT13  024513            2120    4275#
MSLT14  024560            2146    4282#
MSLT15  024607            2168    4294#
MSLT16  024757            2189    4304#
MSLT17  025046            2212    4314#
MSLT2   023704            1794    4206#
MSLT20  025127            2234    4323#
MSLT21  025206            2262    4331#
MSLT22  025242            2291    4336#
MSLT23  025277            2314    4341#
MSLT24  025344            2338    4348#
MSLT25  025404            2395    4354#
MSLT26  025440            2419    4359#
MSLT27  025474            2462    4364#
MSLT3   023763            1829    4214#
MSLT30  025530            2483    4369#
MSLT31  025566            2536    4374#
MSLT32  025622            2683    4379#
MSLT33  025656            2719    4384#
MSLT34  025712            2736    4389#
MSLT35  025746            2758    4394#
MSLT36  026013            2782    4401#
MSLT37  026060            2802    4408#
MSLT4   024050            1925    4223#
MSLT40  026126            2829    4415#
MSLT41  026173            2851    4422#
MSLT42  026242            2901    4429#
MSLT43  026276            2957    4434#
MSLT44  026332            2997    4439#
MSLT45  026412            3042    4447#
MSLT46  026474            3086    4456#
MSLT47  026530            3125    4461#
MSLT5   024132            1934    4232#
MSLT50  026600            3157    4468#
MSLT51  026660            3188    4476#
MSLT6   024201            1973    4239#
MSLT7   024250            1999    4246#
MS57A   023174            1605    4143#
MTINT   017202            1309    1591    3601#
NONSTD  001020            1424#
NOST    016624            1614    3518#
NRZOF   000656            1375#
NXTDRV  002266            1670#   1687
NXTSLV  002340            1679#   1690    *716
OCTP    020010            1575    1584    1618    1627    1636    1646    *720    3275    3304    3455    3459    3465    3751#
```

OCTPE	020000	3293	3297	3322	3324	3326	3328	3330	3332	3334	3336	3338	3340	3749#
		3865												
OCTPE1	020014	3750	3752#											
OCTPG	020166	3761	3764	3771	3775	3780	3785	3787	3792#					
OCTPG0	020204	3793	3795	3798#										
OCTPG1	020210	3756	3799#											
OCTP0	020034	3753	3755	3758#										
OCTP1	020054	3759	3763#											
OCTP2	020062	3762	3765#											
OCTP3	020152	3757	3788#											
OFL	020230	3749*	3751*	3754	3794	3798*	3804#							
PATRN	000776	1415#												
PCNTR	001014	1422#	1656*	1719	1738*	3498								
PEXFL	000732	1397#	3438*	3450	3476*									
PFLG	000662	1377#	3262*	3283*	3308	3350*	3384*	3388	3425*	3430	3434*	3440*	3466*	3471
POST	030570	4807#												
PRE	030446	4766#												
PREFL	000764	1410#												
PSW	000566	1345#	2124*	2128*	3613*									
RBUFF	031324	4850#												
RCDP	001004	1418#												
RDAD	000756	1407#												
RDATA	030176	4689#												
RDRVF	001002	1417#	1655*	3222*										
REGP	015470	3298	3318#	3383	3424									
REGS	000606	1353#	1574	1576	1593									
RTRN	000664	1378#	2122*	3602										
SAV1	000700	1384#												
SAV2	000702	1385#												
SAV3	000704	1386#												
SCF	000726	1395#												
SCHN	001446	1558	1560#											
SCOLP	000706	1387#	1784*	1819*	1844*	1849*	1927*	1960*	1964*	1989*	2013*	2040*	2069*	2075*
		2081*	2104*	2108*	2112*	2130*	2154*	2176*	2198*	2221*	2235*	2263*	2292*	2315*
		2339*	2405*	2444*	2461*	2484*	2537*	2562*	2684*	2720*	2744*	2749*	2768*	2774*
		2787*	2794*	2809*	2820*	2828*	2850*	2902*	2956*	2998*	3043*	3087*	3126*	3158*
		3189*	3393*	3490										
SCOPE	016504	2510	3314	3394	3477	3482#								
SERFL	000766	1411#												
SKAT	001012	1421#	1550*	1737*	1745	1864								
SLAF	000716	1391#												
SLVN	000660	1376#	1626	1628	1670*	1679*	1685*	1686	1688	1862	1863	2686	3533	3556
SLVTYP	001010	1420#	1645	1647	1907	2645								
SN	000540	1328#	1914											
SNPG	020412	3838	3841	3847	3849	3851#								
SNPT	020334	1915	3832#											
SSCF	000720	1392#												
START	001332	1300	1538#											
STATC	001006	1419#	1635	1637	3218									
STATF	001000	1416#												
STATIC	014760	2897	2955	2996	3041	3085	3216#							
STFLG	000734	1398#	1663*	1702	1707*	1747	1866	2143	2165	2186	2209	3216		
STSCD	002516	1666	1707#											
ST0	001662	1596#	1599											
ST1	001704	1602#	1604											
ST2	002202	1304	1654#											

SWR	000570	1346#	1543	1547*	1556*	1664	1698	1704	1734	1749	1769	1868	1916	2141
		2163	2184	2207	3241	3249	3263	3299	3305	3310	3375	3385	3390	3416
		3427	3432	3441	3460	3468	3473	3483	3494	3618	3620*	3627*	3859	3864
		3868												
SWREG	000176	1296#	1547	1556	3618	3627	3859							
TADX	001326	1532#	3520											
TC	000542	1329#	1688*	1885*	1977*	1978	2056*	2095*	2265*	2318*	2342*	2397*	2424*	2486*
		2496*	2502	2544*	2568*	2694*	2739*	2740*	2762*	2765	2771*	2772	2785	2792
		2805*	2806	2812*	2816	2834*	2853*	3130*	3135	3160*	3161*	3191*	3192*	3339
		3533*	3556*	3571*										
TEMP1	000670	1380#	1760	1879	2237*	2241	2254*	3651*	3660	3684*				
TEMP2	000672	1381#	3253*											
TEMP3	000674	1382#	1743*	1755	1771	1774	1862*	1874	1887	3400*	3404*	3407*	3422	3591
		3823	3826											
TEND	002552	1532	1709	1715#	1911									
TENDX	002650	1733	1735	1737#										
TEX	017776	3726	3744#											
TIB	000612	1357#	3654	3658	3663	3668	3678	3681	3688*	3689	3716*	3717*	3718	
TIMER	010576	2553	2576	2611	2614#	2620	2622							
TIMER0	010622	2621#												
TIMER1	010552	2610#	2613											
TIMOK	010670	2559	2582	2638#										
TIMON	010504	2549	2572	2592#										
TIMOVF	010634	2618	2625#											
TINER	017600	3680	3683	3694	3697	3705#								
TKB	000574	1348#	3607	3716										
TKS	000572	1347#	1661*	3713*	3714									
TLAST	001330	1533#	1708	3522*	3523									
TMS1	027305	4532#												
TMS10	027373	2470	4548#											
TMS11	027401	4550#												
TMS12	027407	4552#												
TMS13	027415	4554#												
TMS14	027423	2727	4556#											
TMS15	027431	4558#												
TMS16	027437	4560#												
TMS17	027445	2245	4562#											
TMS18	027453	4564#												
TMS19	027461	2273	4566#											
TMS2	027313	4534#												
TMS20	027467	2301	4568#											
TMS21	027476	2325	4570#											
TMS22	027504	2375	4571#											
TMS23	027513	3062	4573#											
TMS24	027521	4575#												
TMS25	027527	3107	4577#											
TMS26	027535	2974	4579#											
TMS27	027543	4581#												
TMS28	027551	2445	2872	4583#										
TMS29	027557	4585#												
TMS3	027321	2842	2862	4536#										
TMS30	027564	2867	4586#											
TMS31	027572	2406	4587#											
TMS32	027600	2482	4588#											
TMS33	027606	4589#												
TMS33A	027614	2538	4590#											

```
TMS33B  027630            2563    4592#
TMS33C  027643            2653    4594#
TMS33D  027665            2664    4598#
TMS33E  027710            2629    4602#
TMS34   027730            2701    4605#
TMS35   027736            3017    4606#
TMS36   027745            2937    4608#
TMS37   027753            2757    4610#
TMS38   027761            2783    4612#
TMS39   027767            2803    4614#
TMS4    027326            4537#
TMS5    027334            4538#
TMS6    027343            2735    4540#
TMS7    027351            4542#
TMS8    027357            4544#
TMS9    027365            4546#
TOB     000610            1356#   1600    3718*   3724*   3725    3727    3731*   3734*   3738*   3743    3788*   3800*   3810*
                          3813    3818*   3851*   3853*
TOG     017762            3719    3729    3732    3735    3739    3741#   3742    3789    3801    3854
TPB     000600            1350#   3743*   3815*   3817*
TPS     000576            1349#   3741    3811
TREF    000730            1396#
TR00    000622            1361#   1797
TR01    000624            1362#
TR02    000626            1363#
TR03    000630            1364#
TR04    000632            1365#
TR05    000634            1366#
TR06    000636            1367#
TR07    000640            1368#
TR10    000642            1369#
TR11    000644            1370#
TR12    000646            1371#
TR13    000650            1372#
TR14    000652            1373#
TR15    000654            1374#
TSCD    002222            1570    1661#   1706    1739
TSCDA   002416            1668    1691#
TSCD0   002424            1692#   1703
TSCD1   002432            1693#   1714
TSCD2   002460            1698#   1787    1825    1858    1930    1968    1993    2018    2044    2084    2115    2133    2158
                          2180    2202    2225    2258    2287    2310    2335    2391    2415    2455    2478    2531    2586
                          2710    2731    2753    2778    2798    2824    2846    2888    2923    2948    2992    3036    3080
                          3122    3152    3182    3213    3221    3513
TSCD3   002476            1699    1702#
TSTTBL  001056            1448#   1691    1712    3520*   3523*
TTIN    017620            3653    3713#
TTINT   017212            1290    3607#
TTOUT   017662            1564    1567    1573    1582    1606    1616    1625    1634    1644    1718    1754    1873    1895
                          1906    1910    1913    2507    2509    2626    2628    2630    2650    2652    2654    2661    2663
                          2665    2922    3246    3248    3268    3271    3282    3289    3291    3295    3302    3320    3380
                          3382    3421    3423    3446    3449    3453    3457    3463    3511    3588    3590    3623    3666
                          3675    3706    3724#   3730    3740    3863    3867
TTR     017340            1580    1589    1611    1623    1632    1641    1651    1759    1878    3595    3649#   3872
T24FL   000740            1400#   2374*   2378*   3286
UDES    000772            1413#   2896*   2954*   2995*   3040*   3084*   3542    3543*   3547    3548*   3553    3557    3571
```

```
                        3580*
VECT    000604          1352# 1583    1585    1590
WAM     000744          1402#
WBUFF   030712          4848#
WC      000512          1317# 2239*   2345*   2425*   2904*   2925*   2963*   3000*   3045*   3089*   3128*   3162*   3193*
                        3323
WCDP0   001044          1440#
WCDP2   001032          1435#
WCS1    001022          1428#
WCS2    001024          1429#
WDATA   027776          2343  2906    2927    2965    3002    3047    3091    3131    3164    3195    4620#
WDS     001026          1430#
WER     001030          1431#
WMSG27  030376          3288  4757#
WMSG31  030411          3247  4759#
WPGFL   000774          1414# 1662*   1701*
WTAD    000752          1405#
W2FLG   000760          1408#
$DONE   002560          1674  1717#
$ENDAD  002616          1282  1727#
$MNEW   027275          3866  4530#
$MSWR   027265          3862  4528#
$SVPC =  000764         1280# 1285
.      = 031326         1278# 1280    1281#   1283#   1285#   1289#   1295#   1299#   1303#   1308#   1313#   2613#   2620#
                        4849#
.RESTO  020556          3624  3873    3886#
.SAVE   020534          3621  3861    3876#
```

```
$CATCH   1198#   1278
$CHAIN   1198#   1551
$CHNMO   1198#   1667
$RESTO   1198#   3885
$SAVE    1198#   3875
.$ACT1   1198#   1278
.$EOP    1198#   1724


. ABS.  031326     000


 ERRORS DETECTED:  0

 DSKZ:CZTEAB,DSKZ:CZTEAB.SEQ/CRF/SOL=CZTEAB.SML/ML,DSKM:CZTEAB.P11
 RUN-TIME: 6 10 1 SECONDS
 RUN-TIME RATIO: 144/18=7.8
 CORE USED:  11K  (21 PAGES)
```