TM02/TU16

CONTROL LOGIC TEST

MD-11-DZTUC-E

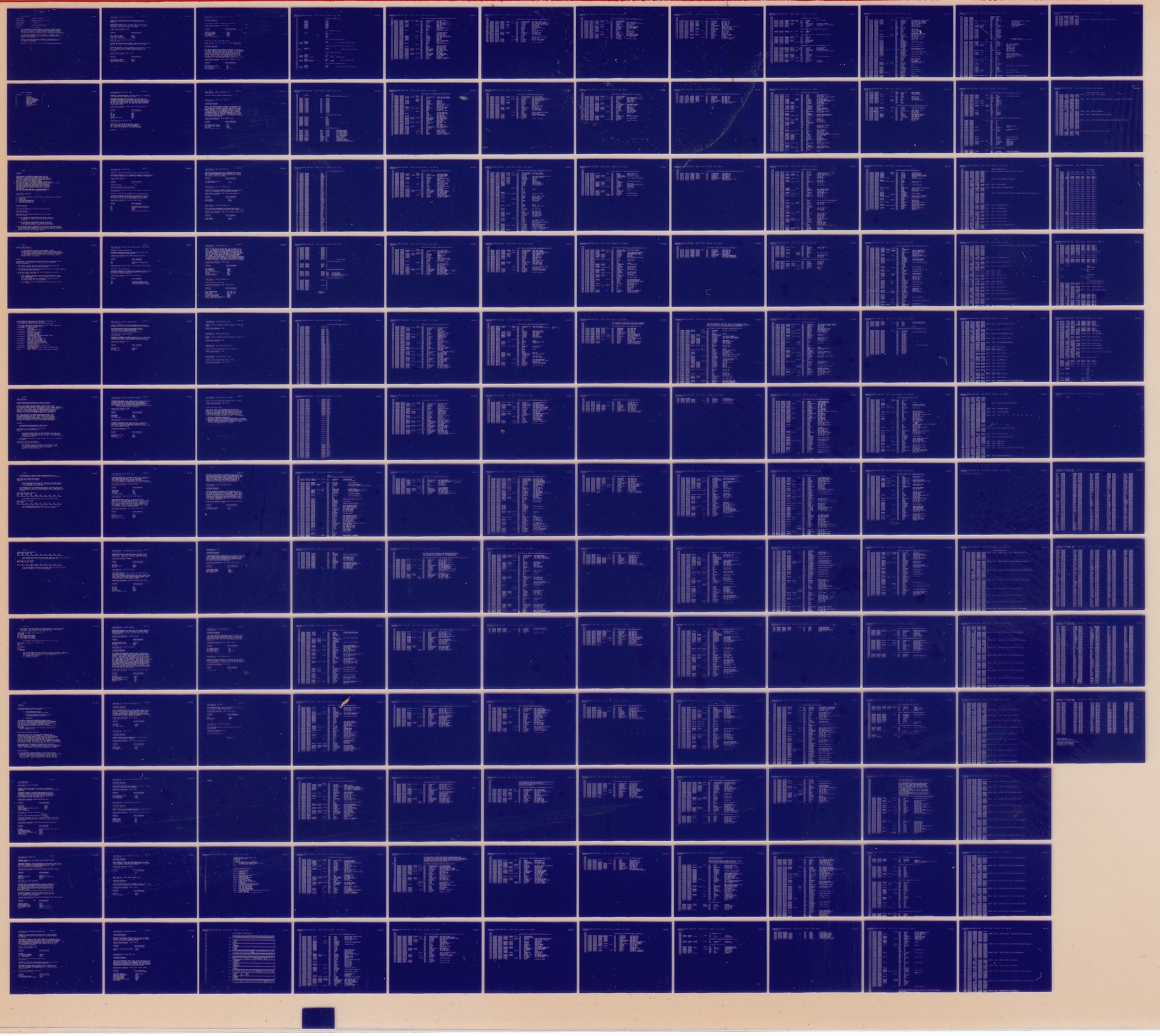
EP-DZTUC-E-DL

FICHE 1 OF 1

JAN 1978

digital

MADE IN USA



IDENTIFICATION

SEQ 0001

PRODUCT CODE: MAINDEC-11-DZTUC-E-D

PRODUCT NAME: TM02/TU16 CONTROL LOGIC TEST

DATE CREATED: 21 APRIL 76

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: R. B. BARNES

TABLE OF CONTENTS

1. ABSTRACT
   ****

THIS PROGRAM IS DESIGNED TO SEQUENTIALLY TEST ALL
CONTROL LOGIC AND DATA FORMATTING WITHIN THE TM02
FORMATTER. EACH TEST WILL ATTEMPT TO ISOLATE FAILURES
TO THE MODULE LEVEL AND PROVIDE PRINTOUT INFORMATION
WHICH WILL IDENTIFY THE FAILING MODULE.
THERE ARE TWO (2) MAJOR AREAS OF TESTING: CONTROL LOGIC
AND DATA FORMATTING. THE CONTROL LOGIC SECTION (TEST 1-41 & 57-64)
WILL TEST ALL ERROR AND STATUS CONDITIONS AS WELL AS
ADDRESSING PROTOCOL AND OPERATIONAL LOGIC SEQUENCES.
THE DATA FORMATTING SECTION (TESTS 42-56) WILL TEST
ALL DATA FORMATS AND TRANSFER PATHS IN ALL POSSIBLE
COMBINATIONS.
THE LEVEL OF FAULT ISOLATION IS POSSIBLE BECAUSE OF
TM02 THE STRUCTURE AND ITS MAINTAINENCE MODES.

2. REQUIREMENTS (HARDWARE)
   ****

A. ANY PDP-11 PROCESSOR - WITH OR WITHOUT A HARDWARE SWITCH REGISTER
B. 8K OF CORE
C. CONSOLE TTY
D. TM02 MAGTAPE CONTROLLER
E. MASSBUS CONTROLLER (RH)
F. TU16 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

***IF THE SOFTWARE SWITCH REGISTER IS USED THEN THE FOLLOWING MESSAGE
   WILL BE TYPED FIRST ; SWR=XXXXXX NEW= THIS WILL ALLOW THE LOADING OF
   THE SOFTWARE SWITCH REGISTER LOC. 176 BEFORE THE TESTING IS STARTED.
   (REFER TO SECTION 5 FOR OPERATOR OPTION)

5. CONSOLE SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

1) TYPE CONTROL G <^G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE ''NEW=" HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:

A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED)
IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

```
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.

***BUT, THE SOFTWARE SWITCH REGISTER CAN ONLY BE  LOADED DYNAMICALLY
   AS STATED ABOVE UNDER CONTROL HEADING.
SW15(100000):   1=HALT ON ERROR
                0=CONTINUE
SW14(040000):   1=LOOP ON ERROR (SCOPE)
                0=CONTINUE
SW13(020000):   1=DO NOT PRINT ERRORS
                0=PRINT ALL ERRORS
SW12(010000):   1=INHIBIT ITERATIONS
                0=ITERATE EACH TEST ITS ASSIGNED AMOUNT
SW11(004000):   1=DO CONTINUOUS CYCLE
                0=HALT AT END OF PASS
SW10(002000):   1=HALT AT END OF CURRENT TEST
                0=CONTINUE TO NEXT TEST
SW9(001000):    1=DO MANUAL ITERVENTION TESTS
                0=INHIBIT MANUAL INTERVENTION
SW8(000400):    1=INHIBIT WRAP AROUND DATA CHECK
                0=DO DATA CHECKS
SW7(000200):    1=INHIBIT WRAP AROUND STATUS CHECK
                0=DO STATUS CHECK
SW6(000100):    1=SELECTABLE WRAP DATA PATTERN (IN SINGLE TEST)
                0=AUTO PATTERN
SW5-0:          SELECT INDIVIDUAL TEST (1-64)** 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 WRAP AROUND 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      C      CS2     DS      ER      AS      MR      TC
004260 000700 033726 000040 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	030000	000001	000000	140307

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

5. THE FOLOWING ARE TWO EXAMPLES OF ERRORS DETECTED BY THE WRAP AROUND DATA TESTS. NOTE THAT EACH WRAP AROUND TEST MAY BE ACCOMPANIED BY EITHER A STATUS ERROR OF A DATA ERROR OR BOTH.

```
LOGIC TEST 42: WRAP 3, NRZ, NORMAL, ODD
BAD STATUS
CS1 EXPT 004270 RCVD 144270
CS2 EXPT 000100 RCVD 000100
DS  EXPT 010600 RCVD 150600
ER  EXPT 000000 RCVD 000100
```

THIS MESSAGE INDICATES BAD STATUS OF VPE (BIT 6 OF ER)

```
LOGIC TEST 44: WRAP 2, NRZ, NORMAL, ODD
BAD DATA
CN:0
G: 11111111
B: 11111011
CN:10
G: 00000000
B: 00001000
```

THIS MESSAGE SHOWS THAT DATA RECEIVED WAS NOT AS EXPECTED. CHARACTER ZERO (CN: 0) SHOWS THAT BIT TWO (2) WAS DROPPED, WHILE CHARACTER TEN (CN: 10) SHOWS BIT THREE (3) HAS BEEN PICKED UP
G: = EXPECTED DATA (GOOD)
B: = ACTUAL DATA (BAD)

7. OPERATION
****

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

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

***REFER TO SECTION 5 FOR DYNAMIC LOADING
OF SOFTWARE SWITCH REGISTER.***

3. PRESS START

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 (SW12=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.
***FOR THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
REFER TO SECTION 5.****

SINGLE TEST SELECTION: (SW0-SW5)

WHEN SW0-SW5 ARE SET TO ZERO (00), THE SCHEDULAR WILL
EXECUTE ALL TESTS (1-64) IN SEQUENCE. IF SW0-SW5 ARE SET
TO SOME SPECIFIC TEST NUMBER (1-64) 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.

WRAP AROUND DATA PATTERNS MAY BE SELECTED VIA SW6 WHEN IN
SINGLE TEST MODE. A TELETYPE REQUEST IS MADE FOR THE DESIRED DATA
PATTERN WHENEVER SWITCH TEN (SW10) AND SWITCH SIX (SW6) ARE SET TO
A ONE (1) WHILE ONE OF THE WRAP TESTS IS SELECTED IN SW0-SW5.

PROGRAM HALTS***

***IF THE SOFTWARE SWITCH REGISTER IS USED AND THE PROGRAM
HALTS THEN THE OPERATOR CAN PRESS A <^G> CONTROL G BEFORE HITTING
CONTINUE. THIS WILL ALLOW THE OPERATOR TO ENTER DATA INTO
LOC. 176 (SWREG). THE FOLLOWING MESSAGE WILL BE TYPED OUT ;
SWR=XXXXXX NEW= (REFER TO SECTION 5 FOR OPERATOR OPTIONS).

8. TEST DESCRIPTION

LOGIC TEST #1: DRIVE ADDRESSING

PURPOSE: VERIFY THE PRESENCE OF TM02 AT THE ADDRESSES SPECIFIED BY THE OPERATOR. TEST OCCURS IMMEDIATELY AFTER DRIVE SELECTION.

PROGRAMMED SEQUENCE: FOR EACH TM02 ADDRESS (0-7) THE C1 REGISTER IS READ, AND THE NON-EXISTANT DRIVE (NED) BIT IS CHECKED. NED IS SET WHEN THE TM02 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: M5984,CABLE,M5983,M8989

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: M5984,CABLE,M5983,M8989,M8985,M8983

CIRCUITS	PRINT REFERENCE
C-LINES	(MB1,2,3),(MBI3),(MBI4),(MBI5)
RH REGISTER SELECT	(BCTA)
TM02 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: M5984,CABLES,M5983YA,M8909,M8905,M8903
••••••••••••••••••••

CIRCUITS	PRINT REFERENCE
C-LINES	(MBI1,2,3)
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 TU16 AND THE SPR BIT.
****
IT IS REQUIRED THAT ONLY ONE SLAVE BE POWERED UP WHEN
THIS TEST IS RUN.
****

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

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

CIRCUITS	PRINT REFERENCE
REGISTER SELECT	(MBI2)
SLAVE ADDRESS BITS	(MR6)
SLAVE ADDRESS LINES	(M8907,2-2),(LAW6)
TU16 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: M8985

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: M8989,M8985

CURCUITS	PRINT REFERENCE
TM02 REGISTER SELECT	(MBI2)
TC FLIP-FLOPS, MULTIPLEXERS	(MR6)

LOGIC TEST #7: FRAME COUNT BIT TEST

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

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

LIKELY FAULT LOCATIONS: M8909

CIRCUITS	PRINT REFERENCE
TM02 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

CIRCUITS	PRINT REFERENCE
TM02 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

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: M8918,SLAVE CABLE, M8903

CIRCUIT	PRINT REFERENCE
MOL	LAW6,TCCM7,M8908,M9081YA,YC
WRL	LAW8,TCCM7,M8908,M9081YA,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,M8983,M8989,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,M8983

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 #28: 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: M8969

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

LOGIC TEST #21: REGISTER MODIFICATION REFUSED

PROGRAMMED SEQUENCE: INIT, SELECT SLAVE AND DRIVE, LOAD 300
0 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 C82, 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	MBI11
CPAR MULTIPLEXER	MBI10

LOGIC TEST #23: FORMAT ERROR (FMT)

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

LIKELY FAULT LOCATIONS: M8905,M8906,M8909

CIRCUIT	PRINT REFERENCE
FORMAT BITS	MR6
ILF DECODE	BF3
ILF FLOP	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, 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, M8903, M8905

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

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, M8989

CIRCUITS	PRINT REFERENCE
REGISTER SELECT LINES	MB1, MB2
REGISTER SELECT DECODE	MBI2
ILR FLOP	MBI11

LOGIC TEST #38: 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: M8989, M8985, M8986, MB CABLES

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

LOGIC TEST 31: OPERATION INCOMPLETE (OPI)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8983, M8989

CIRCUITS	PRINT REFERENCES
OPI TIMER	TCCM5
OPI FLOP	MBI11
OPI TIMER CONTROL	MBI7

LOGIC TEST 32: UNSAFE (UNS)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8989, M8918, 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, M8903

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

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

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

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

CIRCUITS	PRINT REFERENCES
DENSITY BITS	MR6
DENSITY LINES	8DC
PES CIRCUIT	8C3
PES STATUS BIT	TCCM7

LOGIC TEST 35: TAPE CONTROL WRITE (TCW)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATION: M8905

CIRCUIT	PRINT REFERENCES
TCW	MR6

LOGIC TEST 36: FRAME COUNTER STATUS (FCS)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8909, M8903

CIRCUITS	PRINT REFERENCES
FCS BIT	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: M8983, M8911

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

LOGIC TEST 48: PE TAPE MARK (TM)

PROGRAMMED SEQUENCE:

DRIVE IS INITIALIZED, WAM8 IS SET, WRITE TAPE MARK IS SET, AFTER DELAY TAPE MARK BIT IS CHECKED, WAM8 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 M8982.

LIKELY FAULT LOC IONS: M8982, M8981, M8983, M8985

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
WAM8 SIGNAL	MR5

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

PROGRAMMED SEQUENCE:

SAME AS TEST 40 EXCEPT NRZ DENSITY IS SELECTED.

LIKELY FAULT LOCATIONS: M8903, M8904

CIRCUITS	PRINT REFERENCES
WRITE DATA BUFFER	TCCM2
RSD0 MULTIPLEXER	TCCM6
RDA MULTIPLEXERS	TCCM6
TM DETECTOR	CNRZ4
ILLEGAL TAPE MARK FLOP	CNRZ4

SEE NOTE ON PAGE 22 FOR TESTS 42-56

LOGIC TEST 42: WRP3, NRZ, NORMAL, ODD (BIT FIDDLER READ)

PROGRAMMED SEQUENCE:

TAPE CONTROL REGISTER IS LOADED WITH DENSITY 3, FORMAT 14, ODD PARITY WRP3 IS LOADED IN MAINT. REGISTER. READ FUNCTION IS LOADED, EXECUTING WRAP3 CONSISTS LOADING DATA CHARACTERS INTO MAINT. REGISTER DATA FIELD, WHERE THERE ARE MULTIPLEXERS TO BIT FIDDLER, MM CLK IS TOGGLED TO CREATE RDS. THE BIT FIDDLER TRANSMIT DATA ACCESS MASSBUS DATA LINES. WHEN ALL THE DATA HAS BEEN TRANSMITTED AN EOR CLK IS TRANSMITTED TO N REGISTER WHICH BRINGS OPERATION TO A CLOSE.

LIKELY FAULT LOCATIONS: M8906, M8905, MASSBUS P-LINES

CIRCUITS	PRINT REFERENCES
MASSBUS CHAR. ASSEMBLE	BF5
CLK. GENERATOR	BF2
MAINT. REGISTER DATA FIELD	MR2, MR3
RDS GENERATION	MR5

LOGIC TEST 43: WARP3, PE, NORMAL, ODD

JUST LIKE TEST 42 EXCEPT FOR DENSITY BITS.

LOGIC TEST 44: WRAP2, NRZ, NORMAL, ODD

PROGRAMMED SEQUENCE:

WRAP2 IS BIT FIDDLER WRITE, MM CLOCK IS MULTIPLEXED INTO WRT CLK SO THAT IT FORMS WRT STROBE. THE OUTPUT OF THE BIT FIDDLER IS CLOCKED INTO THE DATA FIELD OF THE MAINTENANCE REGISTER. SET UP CONSISTS OF MOVING NRZ, NORMAL FORMAT, ODD PARITY TO UNIT DESCRIPTION MAINT. REGISTER IS LOADED WITH WAM2 WRITE COMMAND IS ISSUED. AFTER THE ACCELERATION DELAY, MM CLOCK ARE GENERATED UNTIL ALL THE DATA HAS BEEN CLOCKED. SEQUENCE IS COMPLETED BY LOADING MANTENANCE REGISTER WITH EOR CLR. THE SEQUENCE IS REPEATED WITH VARYING DATA PATTERNS.

LIKELY FAULT LOCATIONS: M8986, M8985, M8983

CIRCUITS	PRINT REFERENCES
BIT FIDDLER CHAR UNPACK	BF4
BIT FIDDLER DATA REQUEST	BF2
WRT STRB.	TCCM4
MAINT. REG. DATA FIELD	MR2, MR3

LOGIC TEST 45: WRP2, PE, NORMAL, ODD

THE TEST IS EXACTLY LIKE TEST 44 EXCEPT THAT PE WRT CLK ENBL L MUST BE ASSERTED BY M8902 TO ENABLE WR TO STROBE. THIS DOES NOT HAPPEN UNTIL THE PE WRITE CONTROL CIRCUIT HAS CLOCKED THROUGH THE PREAMBLE.

CIRCUITS	PRINT REFERENCES
(IN ADDITION TO TEST 44)	
PE WRITE CONTROL	TCPE3

LOGIC TEST 46: WRP1, NRZ, NORMAL, ODD

THIS TEST IS EXACTLY LIKE TEST 44 EXCEPT THE WRITE BUFFER (TCCM2) IS MULTIPLEXED TO THE MAINTENANCE REGISTER.

LIKELY FAULT LOCATIONS: M8903, M8904 (CRC GENERATOR)

CIRCUITS	PRINT REFERENCES
WRITE BUFFER	TCCM2
CRC GENERATOR	CNRZ2

LOGIC TEST 47: WRAP1, PE, NORMAL, ODD

IN PE MODE BOTH THE PREAMBLE AND POSAMBLE ARE CLOCKED THROUGH THE WRITE BUFFER IN ADDITION TO PHASE ENCODED DATA.

LIKELY FAULT LOCATIONS: M8902 (WRITE CONTROL STATES), M8903

CIRCUITS	PRINT REFERENCE
WRITE BUFFER	TCCM2
WRITE CONTROL	TCPE3

LOGIC TEST 50: WRAP0, NORMAL, ODD

WRAP 0 IS THE MOST COMPLETE OF THE TM02 WRAPAROUND DATA PATH. IT CONSISTS OF A WRITE OPERATION IN WHICH THE OUTPUT OF THE WRITE DATA BUFFER IS MULTIPLEXED TO THE READ DATA INPUTS, CHECKED AND LOADED INTO THE MAINTENANCE REGISTER FOR RETRIEVAL BY THE PROCESSOR. THE WHOLE OPERATION USES THE TYPE SYSTEM CLOCKS AND HAPPENS AT THE PROPER DATA RATES. MM CLK SERVES AS A FLAG ANNOUNCING WHEN A NEW CHARACTER HAS BEEN LOADED INTO THE MAINTENANCE REGISTER. IN PE MODE EVERY OTHER CHARACTER IS READ TO ALLOW SUFFICIENT PROCESSOR LOOP TIME. IN NRZ WRAP 0 IS EXPECTED TO PRODUCE LRC ERRORS BECAUSE THE TM02 DOES NOT WRITE THE LRC CHARACTER.

LIKELY FAULT LOCATIONS: M8904, M8903

CIRCUITS	PRINT REFERENCES
CRC GENERATOR	CNRZ2
CRC CHECKOUT	CNRZ3
CRC, CRC STROBE	TCCM4
READ LINE MULTIPLEXERS	TCCM6
MM CLK	MR5
CRC READ TIMING	CNRZ4
SHUTDOWN CIRCUITRY	TCCM5

LOGIC TEST 51: WRP0, PE, NORMAL, ODD

REPEAT OF TEST 50 IN PE MODE.

LIKELY FAULT LOCATIONS: M8901, M8902, M8903

CIRCUITS	PRINT REFERENCES
DATA DISCRIMINATOR	D82, D84, D86
PHASE LOCKED CLOCK	D83, D85, D87
SKEW BUFFER	D83, D85, D87
PE WRITE MAJOR STATES	TCPE3
PE READ MAJOR STATES	TCPE5
WRAP 0 CIRCUIT TO BLOCK RLT RDS	TCPE3
DESKEW BUFFER READ COUNTER	TCPE4

LOGIC TEST 52: CORE DUMP WRITE, WAM2

REPEAT OF TEST 44 EXCEPT BIT FIDDLER OPERATES IN CORE DUMP MODE;

LIKELY FAULT LOCATION: M8906

LOGIC TEST 53: CORE DUMP READ, WAM3

REPEAT OF TEST 42 EXCEPT BIT FIDDLER OPERATES IN CORE DUMP MODE;

LIKELY FAULT LOCATION: M8906

LOGIC TEST 54: EVEN PARITY WRITE - WAM1

REPEAT OF TEST 46 EXCEPT EVEN PARITY IS SPECIFIED.

LIKELY FAULT LOCATION: M8903

LOGIC TEST 55: EVEN PARITY READ: WAM0,

REPEAT OF TEST 50 EXCEPT EVEN PARITY IS USED,

LIKELY FAULT LOCATIONS: M8903, M8904

LOGIC TEST 56: READ REVERSE, WAM3 (M8906)
••••••••••••

REPEAT OF TEST 42 EXCEPT READ REVERSE COMMAND IS ISSUED.

LIKELY FAULT LOCATIONS: M8908, M8909
••••••••••••••••••••••

NOTE: FOR TESTS 42-56

FOR THE MOST PART, THIS DIAGNOSTIC TESTS PARTICULAR
AREAS OF THE TM02 LOGIC INDEPENDENT OF THE TU16. HOWEVER
THERE ARE A FEW SIGNALS WHICH ARE REQUIRED FROM THE TU16
TO COMPLETE THE TESTS, AND AT LEAST ONE CASE WHERE TU16
FAILURES INTERFERE WITH THE TESTS. THE KNOWN CASES ARE
LISTED HERE AND SHOULD BE CHECKED AS PART OF THE DEBUGGING.

1. MOL(SB)L: REQUIRED TO ENABLE CLOCK
2. CLOCK(SB)L: REQUIRED TO GENERATE ACCELERATION AND SHUTDOWN.
3. WRITE CLOCK(SB)L: USED IN WAM0 TO GENERATE DATA AND REC(SB)L
4. RED0(SB)L: SHOULD NOT OCCUR DURING WRAP AROUND TESTS, BUT WILL
   INTERFERE WITH THEIR OPERATION IF CAUSED BY A FAILURE
   SUCH AS A GROUNDED OUTPUT FROM THE G056.

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 57: CYCLIC REDUNDANCY ERROR

PROGRAMMED SEQUENCE:

FIRST THE DIAGNOSTIC PERFORMS A WRAP8 DESIGNED TO LOAD THE CRC CHECKER IN A KNOWN MANNER. CHECK ARE MADE FOR LRC ERROR AND THE CONTENT OF CRC REGISTER. THEN A WRITE OPERATION IS PERFORMED USING A MAINT. MODE (IICC) WHICH INHIBITS THE INITIALIZATION OF THE CRC CHECKER. THE CRC CHECKER LOGIC WHICH HAS NOT BEEN CLEARED SHOULD DETECT A CRC ERROR. UNEXPECTED ERROR BITS MAY INDICATE PROBLEMS WITH THE WRITE OPERATION.

LIKELY FAULT LOCATIONS: M8905, M8904, G856, SLAVE CABLE, M8918

CIRCUITS	PRINT REFERENCES
MM FUNCTION DECODE	MR5
CRC CHECK CIRCUIT	CNR23

LOGIC TEST 60: LRC SEQ 0034

PROGRAMMED SEQUENCE:

A WRITE OPERATION IS PERFORMED WITH A MM FUNCTION (INC TMRL) WHICH ASSERTS WD(8B) SL THROUGHOUT THE RECORD. ALL ONES DATA IS USED SO THAT THE FUNCTION ONLY INTERFERES WITH THE WRITING OF THE LRC CHARACTER WHEN NONE OF THE TM82 WRITE DATA LINES SHOULD BE ASSERTED.

LIKELY FAULT LOCATIONS: M8585, M8903, M8910, M8984

CIRCUITS	PRINT REFERENCES
MM FUNCTION DECODE	MR8
WRITE LINE DRIVERS	TCCM2
WRITE HEAD DRIVERS	LAW3, 4
LRC CHECKING	CHR23

LOGIC TEST 61: PE CORRECTABLE DATA

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905, M8901, M8902

CIRCUITS	PRINT REFERENCES
MM FUNCTION DECODE	MR5
BIT STROBE CIRCUIT	D84
DEAD TRACK FLOP	D85, TCPE2
DEAD TRACK REGISTER	MR4

LOGIC TEST 62: PE INCORRECTABLE DATA

REPEAT OF TEST 61, EXCEPT THAT THE MAINT, MODE FUNCTION GROUNDS BITS STROBE FOR BITS 1, 2 AND THE WD LINE FOR BIT 5 IN HELD ASSERTED, INC, DATA AND PCF ERRORS ARE EXPECTED,

LIKELY FAULT LOCATIONS: M8902, M8901

CIRCUIT	PRINT REFERENCE
INC ERROR, PEF,	TCPE2

LOGIC TEST 63: PE FORMAT
••••••••••••

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

LIKELY FAULT LOCATIONS: M8902, M8903, M8905
••••••••••••••••••••••

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

LOGIC TEST 64: FRAME COUNT OVERFLOW
••••••••••••

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

LIKELY FAULT LOCATION: M8909
••••••••••••••••••••••

FRAME COUNT REGISTER MBI8

9. LISTING

DZTUCE,P11

```
                        .TITLE TM02/TU16 CONTROL LOGIC TEST
                        ;MAINDEC-11-DZTUC-E-D
                        ;21 OCT 75
                        ;R. BARNES
                        ;REVISED 21 APRIL 76 BY S. CARPENTER
                        ;       TO SUPPORT THE DYNAMIC LOADING OF THE
                        ;       SOFTWARE SWITCH REGISTER
                        .ABS


                        ;CONSOLE SWITCHES**********
                        ;
                        ;SW15:  1=HALT ON ERROR
                        ;       0=CONTINUE
                        ;SW14:  1=LOOP ON ERROR
                        ;       0=CONTINUE
                        ;SW13:  1=DO NOT PRINT ERRORS
                        ;       0=PRINT ERRORS
                        ;SW12:  1=INHIBIT ITERATIONS
                        ;       0=DO ITERATIONS
                        ;SW11:  1=CONTINUOUS CYCLE
                        ;       0=HALT AT END OF PASS
                        ;SW10:  1=HALT AT END OF EACH TEST
                        ;       0=CONTINUE
                        ;SW9:   11=DO MANUAL INTERVENTION TESTS
                        ;       0=INHIBIT MANUAL INTERVENTION
                        ;SW8:   1=NO WRAP DATA CHECK
                        ;       0=DO WRAP DATA CHECK
                        ;SW7:   1=NO WRAP STATUS CHECK
                        ;       0=DO WRAP STATUS CHECK
                        ;SW6:   1=SELECTABLE WRAP DATA PATTERN (IN SINGLE TEST)
                        ;       0=AUTO PATTERNS
                        ;SW0-5: SELECT TEST NUMBER ;; 00=ALL TESTS
```

```
                        ;TU16 REGISTER BITS**********

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

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

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

                                    ;TRAP CATCHERS**********

            000000                  .=0
                                    .REPT 200
                                    .+2
                                    HALT
                                    .ENDR

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

            000060                  .=60
000060  023602                      TTINT           ;TTY INTERRUPT HEADER ADDRESS
000062  000000                      0

                                    ;SOFTWARE SWITCH REGISTER LOC, 176**********

            000176                  .=176
000176  000000                      SWREG:  0               ;SOFTWARE SWITCH REGISTER


                                    ;START ADDRESS**********

            000200                  .=200
000200  005000                      CLR     R0
000202  000167  001206              JMP     START   ;PROGRAM START

            000210                  .=210
000210  012700  000001              MOV     #1,R0           ;SET NO HEADER FLAG
000214  000167  001174              JMP     START

                                    ;TM02 INTERRUPT VECTOR**********

            000224                  .=224
000224  023570                      MTINT           ;TAPE INTERRUPT HANDLER ADDRESS
000226  000340                      340
```

```
                        .=510
                        ;MASS BUS REGISTER EQUIVS**********

000510  172440    C1:     172440
000512  172442    WC:     172442
000514  172444    BA:     172444
000516  172446    FC:     172446
000520  172450    CS:     172450
000522  172452    DS:     172452
000524  172454    ER:     172454
000526  172456    AS:     172456
000530  172460    CC:     172460
000532  172462    DB:     172462
000534  172464    MR:     172464
000536  172466    DT:     172466
000540  172470    SN:     172470
000542  172472    TC:     172472

                        ;ILLEGAL FUNCTION CODES

000544  005405    ILFT:   5405
000546  007415            7415
000550  016423            16423
000552  020437            20437
000554  022443            22443
000556  025447            25447
000560  031455            31455
000562  033465            33465
000564  036473            36473

                        ;CONSTANTS**********

000566  177776    PSW:    177776          ;PROCESSOR STATUS
000570  177570    SWR:    177570          ;SWITCH REGISTER
000572  177560    TKS:    177560          ;TTY READER STATUS
000574  177562    TKB:    177562          ;TTY READ BUFFER
000576  177564    TPS:    177564          ;TTY PUNCH STATUS
000600  177566    TPB:    177566          ;TTY PUNCH BUFFER
000602  177777    SERNUM: 177777          ;SERIAL NUMBER
000604  000011    DRVTP:  011             ;DRIVE TYPE
000606  000020    ITAMT:  20              ;ITERATION AMOUNT
000610  000224    VECT:   224             ;INTERRUPT VECTOR(RH)
000612  172440    REGS:   172440          ;STARTING REGISTER ADDRESS
```

```
                                        ;FLAGS AND COUNTERS**********
                
000614  000000          TOB:    0
000616  000000          TIB:    0
000620  000000          HDRFL:  0
000622  000000          EMADDR: 0
000624  000000          DRVN:   0
000626  000000          TR00:   0
000630  000000          TR01:   0
000632  000000          TR02:   0
000634  000000          TR03:   0
000636  000000          TR04:   0
000640  000000          TR05:   0
000642  000000          TR06:   0
000644  000000          TR07:   0
000646  000000          TR10:   0
000650  000000          TR11:   0
000652  000000          TR12:   0
000654  000000          TR13:   0
000656  000000          TR14:   0
000660  000000          TR15:   0
000662  000000          NRZOF:  0
000664  000000          SLVN:   0
000666  000000          PFLG:   0
000670  000000          RTRN:   0
000672  000000          ERADD:  0
000674  000000          TEMP1:  0
000676  000000          TEMP2:  0
000700  000000          TEMP3:  0
000702  000000          ITCNT:  0
000704  000000          SAV1:   0
000706  000000          SAV2:   0
000710  000000          SAV3:   0
000712  000000          SCOLP:  0
000714  000000          ITRLP:  0
000716  000000          EXFL:   0
000720  000000          ATAF:   0
000722  000000          SLAF:   0
000724  000000          SSCF:   0
000726  000000          ERRF:   0
000730  000000          ASF:    0
000732  000000          SCF:    0
000734  000000          TREF:   0
000736  000000          PEXFL:  0
000740  000000          STFLG:  0
000742  000000          LTADD:  0
000744  000000          T24FL:  0
000746  000000          ADDFL:  0
000750  000000          WAM:    0
000752  000000          FUN:    0
000754  000000          DATC:   0
000756  000000          WTAD:   0
000760  000000          DATAD:  0
000762  000000          RDAD:   0
000764  000000          W2FLG:  0
000766  000000          DERFL:  0
```

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

                                        ;EXPT WRAP STATUS********

                                WC81:   0
                                WC82:   0
                                WDS:    0
                                WER:    0

                                        ;DATA PATTERN GENERATORS******

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

                                        ;CORE DUMP PATTERNS********

                                WCDP2:  5
                                        5
                                        12
                                        12
                                        0
                                WCDP8:  17
                                        17
                                        17
                                        17
                                        0
```

```
                                        ;LOGIC TEST ENTRY TABLE**********

001066  000000          T8TTBL: 0
001070  000000                  0
001072  002310          T1AD:   LT1
001074  002310          T1IAD:  LT1
001076  002540          T2AD:   LT2
001100  002540          T2IAD:  LT2
001102  002746          T3AD:   LT3
001104  002750          T3IAD:  LT3IT
001106  003132          T4AD:   LT4
001110  003132          T4IAD:  LT4
001112  003410          T5AD:   LT5
001114  003416          T5IAD:  LT5IT
001116  003612          T6AD:   LT6
001120  003614          T6IAD:  LT6IT
001122  004024          T7AD:   LT7
001124  004026          T7IAD:  LT7IT
001126  004150          T10AD:  LT10
001130  004152          T10IAD: LT10IT
001132  004304          T11AD:  LT11
001134  004306          T11IAD: LT11IT
001136  004546          T12AD:  LT12
001140  004550          T12IAD: LT12IT
001142  004754          T13AD:  LT13
001144  004764          T13IAD: LT13IT
001146  005056          T14AD:  LT14
001150  005116          T14IAD: LT14IT
001152  005166          T15AD:  LT15
001154  005226          T15IAD: LT15IT
001156  005276          T16AD:  LT16
001160  005336          T16IAD: LT16IT
001162  005406          T17AD:  LT17
001164  005446          T17IAD: LT17IT
001166  005516          T20AD:  LT20
001170  005532          T20IAD: LT20IT
001172  005662          T21AD:  LT21
001174  005676          T21IAD: LT21IT
001176  006026          T22AD:  LT22
001200  006042          T22IAD: LT22IT
001202  006152          T23AD:  LT23
001204  006166          T23IAD: LT23IT
001206  006302          T24AD:  LT24
001210  006316          T24IAD: LT24IT
001212  006626          T25AD:  LT25
001214  006642          T25IAD: LT25IT
001216  006760          T26AD:  LT26
001220  006766          T26IAD: LT26IT
001222  007174          T27AD:  LT27
001224  007220          T27IAD: LT27IT
001226  007312          T30AD:  LT30
001230  007334          T30IAD: LT30IT
001232  007630          T31AD:  LT31
001234  007644          T31IAD: LT31IT
001236  010160          T32AD:  LT32
```

```
001240  010174          T32IAD: LT32IT
001242  010312          T33AD:  LT33
001244  010326          T33IAD: LT33IT
001246  010414          T34AD:  LT34
001250  010436          T34IAD: LT34IT
001252  010556          T35AD:  LT35
001254  010572          T35IAD: LT35IT
001256  010730          T36AD:  LT36
001260  010744          T36IAD: LT36IT
001262  011050          T37AD:  LT37
001264  011064          T37IAD: LT37IT
001266  011220          T40AD:  LT40
001270  011242          T40IAD: LT40IT
001272  011346          T41AD:  LT41
001274  011362          T41IAD: LT41IT
001276  011614          T42AD:  LT42
001300  011614                  LT42
001302  011726          T43AD:  LT43
001304  011726                  LT43
001306  012010          T44AD:  LT44
001310  012010                  LT44
001312  012116          T45AD:  LT45
001314  012116                  LT45
001316  012200          T46AD:  LT46
001320  012200                  LT46
001322  012306          T47AD:  LT47
001324  012306                  LT47
001326  012374          T50AD:  LT50
001330  012374                  LT50
001332  012502          T51AD:  LT51
001334  012502                  LT51
001336  012564          T52AD:  LT52
001340  012564                  LT52
001342  012676          T53AD:  LT53
001344  012676                  LT53
001346  013024          T54AD:  LT54
001350  013024                  LT54
001352  013074          T55AD:  LT55
001354  013074                  LT55
001356  013144          T56AD:  LT56
001360  013144                  LT56
001362  013222          T57AD:  LT57
001364  013252          T57IAD: LT57IT
001366  013562          T60AD:  LT60
001370  013600          T60IAD: LT60IT
001372  014006          T61AD:  LT61
001374  014034          T61IAD: LT61IT
001376  014264          T62AD:  LT62
001400  014312          T62IAD: LT62IT
001402  014536          T63AD:  LT63
001404  014564          T63IAD: LT63IT
001406  015000          T64AD:  LT64
001410  015014          T64IAD: LT64IT
001412  000000          TADX:   0
```

```
                                        .EVEN
                                        ;PROGRAM START AND HOUSEKEEPING**********
001414  012777  000340  177144  START:  MOV     #340,@PSW       ;SET PRIORITY
001422  012706  000500                  MOV     #500,SP         ;SET STACK POINTER

001426  013746  000006          SUBWR:  MOV     @6,-(SP)                ;SAVE VECTORS
001432  013746  000004                  MOV     @4,-(SP)
001436  012737  001456  000004          MOV     #18,@4                  ;SET UP FOR TIMEOUT
001444  022777  177777  177116          CMP     #-1,@SWR                ;REFERENCE HARDWARE SWITCH REGISTER
001452  001402                          BEQ     2$
001454  000404                          BR      3$
001456  022626                  1$:     CMP     (SP)+,(SP)+             ;ADJUST STACK
001460  012767  000176  177102  2$:     MOV     #SWREG,SWR              ;POINT TO SOFTWARE SWITCH REG
001466  012637  000004          3$:     MOV     (SP)+,@4                ;RESTORE VECTORS
001472  012637  000006                  MOV     (SP)+,@6
001476  022767  000176  177064          CMP     #SWREG,SWR              ;IS SWREG SELECTED
001504  001002                          BNE     4$
001506  004767  023002                  JSR     PC,CNTLU                ;CHECK FOR CONTROL G
001512  005700                  4$:     TST     R0
001514  001136                          BNE     ST2
001516  005067  177272                  CLR     SKAT            ;CLEAR SKIP ADDRESS TEST FLAG
001522  012704  025172                  MOV     #MSG1,R4
001526  004767  022312                  JSR     PC,TTOUT        ;PRINT TITLE
001532  012704  027004                  MOV     #MSG44,R4
001536  004767  022302                  JSR     PC,TTOUT        ;REQUEST REGISTER ADDRESS
001542  016703  177044                  MOV     REGS,R3
001546  004767  022434                  JSR     PC,OCTP         ;PRINT CURRENT ADDRESS
001552  012705  000612                  MOV     #REGS,R5        ;SET ADDRESS SAVE LOC
001556  012701  000006                  MOV     #6,R1           ;SET SIZE OF RESPONSE
001562  012702  176480                  MOV     #176480,R2      ;SET UPPER LIMIT
001566  012703  172300                  MOV     #172300,R3      ;SET LOWER LIMIT
001572  004767  022010                  JSR     PC,TTR          ;GO GET RESPONSE
001576  012704  027026                  MOV     #MSG45,R4
001602  004767  022236                  JSR     PC,TTOUT        ;REQUEST VECTOR
001606  016703  176776                  MOV     VECT,R3
001612  004767  022370                  JSR     PC,OCTP         ;PRINT CURRENT VECTOR
001616  012705  000610                  MOV     #VECT,R5        ;SET ADDRESS SAVE LOC
001622  012701  000003                  MOV     #3,R1           ;SET SIZE OF RESPONSE
001626  012702  000224                  MOV     #224,R2         ;SET UPPER LIMIT
001632  012703  000150                  MOV     #150,R3         ;SET LOWER LIMIT
001636  004767  021744                  JSR     PC,TTR          ;GO GET RESPONSE
001642  016700  176742                  MOV     VECT,R0         ;GET VECTOR
001646  012720  023570                  MOV     #MTINT,(R0)+    ;LOAD INTERRUPT ADDRESS IN VECTOR
001652  012710  000340                  MOV     #340,(R0)       ;LOAD PRIORITY
001656  016700  176730                  MOV     REGS,R0         ;GET START OF REGS
001662  012701  000016                  MOV     #16,R1          ;SET NUMBER OF REGS
001666  012702  000510                  MOV     #C1,R2          ;GET START OF TABLE
001672  010022          ST0:    MOV     R0,(R2)+        ;BUILD TABLE
001674  062700  000002                  ADD     #2,R0           ;BUMP ADDRESS
001700  005301                          DEC     R1              ;SEE IF DONE
001702  001373                          BNE     ST0             ;IF NOT; BR
001704  012702  000614                  MOV     #TOB,R2
001710  012700  000077                  MOV     #77,R0
001714  005022          ST1:    CLR     (R2)+           ;CLEAR FLAGS + COUNTERS
001716  005300                  DEC     R0
```

```
   433  001720  001375                    BNE     ST1
   434  001722  012704  027254            MOV     #MSG52,R4
   435  001726  004767  022112            JSR     PC,TTOUT        ;PRINT NRI ONLY REQUEST
   436  001732  012705  000662            MOV     #NRZOF,R5
   437  001736  012701  000001            MOV     #1,R1           ;SET SIZE OF ENTRY
   438  001742  012702  000001            MOV     #1,R2           ;SET UPPER LIMIT
   439  001746  012703  000000            MOV     #0,R3           ;SET LOWER LIMIT
   440  001752  004767  021630            JSR     PC,TTR          ;GO GET RESPONSE
   441  001756  012704  027363            MOV     #MSG56,R4
   442  001762  004767  022056            JSR     PC,TTOUT        ;REQUEST STATIC ONLY
   443  001766  012705  001012            MOV     #STATC,R5       ;SET ADDRESS OF STATIC FLAG
   444  001772  012701  000001            MOV     #1,R1           ;SET SIZE OF RESPONSE
   445  001776  012702  000001            MOV     #1,R2           ;SET UPPER LIMIT
   446  002002  012703  000000            MOV     #0,R3           ;SET LOWER LIMIT
   447  002006  004767  021574            JSR     PC,TTR          ;GET RESPONSE
   448  002012  005067  177000    ST2:    CLR     PCNTR           ;CLEAR PASS COUNTER
```

```
                                        ;TEST SCHEDULAR**********
002016  005067  176756        TSCD:   CLR     WPGFL           ;CLEAR WRAP PATRN FLAG
002022  005067  176712                CLR     STFLG           ;CLEAR SINGLE TEST FLAG
002026  017700  176536                MOV     @SWR,R0
002032  042700  177700                BIC     #177700,R0
002036  005700                        TST     R0
002040  001051                        BNE     STSCD           ;GO SELECT SINGLE TEST
002042  012767  001066  176672        MOV     #TSTTBL,LTADD
002050  062767  000004  176664 TSCD0: ADD     #4,LTADD
002056  016767  176660  176630        MOV     LTADD,ITRLP
002064  062767  000002  176622        ADD     #2,ITRLP        ;SET ITERATION ADDRESS
002072  005777  176644                TST     @LTADD
002076  001002                        BNE     TSCD1
002100  000167  000132                JMP     TEND            ;GO TO END ROUTINE
002104  005067  176510        TSCD1:  CLR     HDRFL           ;CLEAR PRINT HEADER FLAG
002110  017700  176626                MOV     @LTADD,R0       ;SET POINTER TO TEST
002114  000110                        JMP     (R0)            ;GO TO TEST
002116  004767  022320        TSCD2:  JSR     PC,CKSWR        ;CHECK FOR CNTL G
002122  032777  002000  176440        BIT     #2000,@SWR      ;SEE IF HALT ON TEST
002130  001404                        BEQ     TSCD3           ;IF NOT; BR
002132  004767  023024                JSR     PC,STOP
002136  005067  176636                CLR     WPGFL           ;CLEAR WRAP DATA GENERATOR FLAG
002142  005767  176572        TSCD3:  TST     STFLG           ;SE IF SINGLE TEST
002146  001740                        BEQ     TSCD0           ;IF NOT; BR
002150  017700  176414                MOV     @SWR,R0
002154  042700  177700                BIC     #177700,R0      ;MASK TEST NUMBER
002160  005700                        TST     R0              ;SEE IF RETURN TO ALL
002162  001715                        BEQ     TSCD            ;IF SO; BR
002164  012767  000001  176546 STSCD: MOV     #1,STFLG        ;SET SINGLE TEST FLAG
002172  022700  000065                CMP     #65,R0          ;SEE IF EXCEEDED TESTS
002176  003417                        BLE     TEND            ;IF SO; BR
002200  000241                        CLC
002202  006100                        ROL     R0
002204  006100                        ROL     R0              ;SET TABLE MODIFIER
002206  012767  001066  176526        MOV     #TSTTBL,LTADD
002214  060067  176522                ADD     R0,LTADD        ;SET TEST POINTER
002220  016767  176516  176466        MOV     LTADD,ITRLP
002226  062767  000002  176460        ADD     #2,ITRLP        ;SET ITERATION POINTER
002234  000723                        BR      TSCD1
002236  012704  026644        TEND:   MOV     #MSG41,R4
002242  004767  021576                JSR     PC,TTOUT        ;PRINT END OF PASS
002246  016703  176544                MOV     PCNTR,R3
002252  004767  021730                JSR     PC,OCTP         ;PRINT PASS NUMBER
002256  032777  004000  176304        BIT     #4000,@SWR      ;SEE IF HALT ON PASS
002264  001002                        BNE     TENDX           ;IF NOT; BR
002266  004767  022670                JSR     PC,STOP
002272  012767  000001  176514 TENDX: MOV     #1,SKAT         ;SET SKIP ADDRESS TEST FLAG
002300  005267  176512                INC     PCNTR           ;BUMP PASS COUNTER
002304  000167  177506                JMP     TSCD            ;RESTART
```

```
                                                ;LOGIC TEST 11 DRIVE ASSESSING**********
002310  005767  176500          LT11:   TST     $KAT            ;SEE IF SKIP ADDRESS TESTS
002314  001403                          BEQ     LT1G0           ;IF NOT: BR
002316  005767  176416                  TST     $TFLG           ;SEE IF SINGLE TEST
002322  001504                          BEQ     LT1X            ;IF NOT: BR
002324  012704  025315          LT1G0:  MOV     #MSG2A,R4
002330  004767  021510                  JSR     PC,TTOUT        ;PRINT TEST INSTRUCTIONS
002334  012767  027410  176260  LT1G:   MOV     #MSLT1,EMADDR   ;SET HEADER ADDRESS
002342  012704  025276                  MOV     #MSG2,R4
002346  004767  021472                  JSR     PC,TTOUT        ;REQUEST DRIVE NUMBER
002352  012705  000624                  MOV     #DRVN,R5
002356  012701  000001                  MOV     #1,R1
002362  012702  000007                  MOV     #7,R2
002366  012703  000000                  MOV     #0,R3
002372  004767  021210                  JSR     PC,TTR          ;GET DRIVE NUMBER
002376  005767  176272                  TST     TEMP1           ;SEE IF ANOTHER DRIVE
002402  001454                          BEQ     LT1X            ;IF NOT: BR
002404  005001                          CLR     R1              ;SELECT DRIVE 0
002406  012700  000010                  MOV     #10,R0          ;SET NUMBER OF DRIVES
002412  012777  000040  176100  LT1A:   MOV     #40,@CS         ;INIT
002420  010177  176074                  MOV     R1,@CS          ;SELECT DRIVE
002424  005777  176060                  TST     @CS             ;ACCESS DRIVE
002430  032777  010000  176062          BIT     #10000,@CS      ;SEE IF NED
002436  001005                          BNE     LT1B            ;IF SO: BR
002440  026701  176160                  CMP     DRVN,R1         ;SEE IF SHOULD BE NED
002444  001407                          BEQ     LT1C            ;IF NOT: BR
002446  000167  000022                  JMP     LT1ER           ;ELSE GO TO ERROR
002452  026701  176146          LT1B:   CMP     DRVN,R1         ;SEE IF SHOULD BE NED
002456  001002                          BNE     LT1C            ;IF SO: BR
002460  000167  000020                  JMP     LT1ER1          ;ELSE GO TO ERROR
002464  005300                  LT1C:   DEC     R0
002466  001722                          BEQ     LT1G            ;IF DONE ALL: BR
002470  005201                          INC     R1              ;SELECT NEXT DRIVE
002472  000747                          BR      LT1A            ;CONTINUE
002474  012767  000001  176214  LT1ER:  MOV     #1,EXFL         ;FLAG EXPT
002502  000403                          BR      LT1ER2
002504  012767  000002  176204  LT1ER1: MOV     #2,EXFL         ;FLAG NOT EXPT
002512  012767  025444  176152  LT1ER2: MOV     #MSG3,ERADD     ;FLAG CONDITION
002520  012767  002412  176164          MOV     #LT1A,SCOLP     ;SET SCOPE ADDRESS
002526  004767  017262                  JSR     PC,LTGER        ;GO PRINT LOGIC TEST ERROR
002532  000754                          BR      LT1C            ;CONTINUE TEST
002534  000167  177356          LT1X:   JMP     TSCD2           ;RETURN TO SCHED
```

```
                                        ;LOGIC TEST 2: REGISTER ADDRESSING**********
                                LT2:    NOP
002540  000240                          
002542  012777  000040  175750  LT2IT:  MOV     #40,@CS         ;INIT
002550  016777  176050  175742          MOV     DRVN,@CS        ;SELECT DRIVE
002556  012767  027464  176036          MOV     #MSLT2,EMADDR   ;SAVE LT2 HEADER ADDRESS
002564  012705  000510                  MOV     #C1,R5          ;SET ADDRESS OF FIRST REGISTER
002570  012700  000016                  MOV     #16,R0          ;SET NUMBER OF REGISTERS
002574  012702  000626                  MOV     #TR00,R2        ;SET START OF REGISTER BUFFER
002600  011501                  LT2A:   MOV     (R5),R1
002602  011112                          MOV     (R1),(R2)       ;READ REGISTER
002604  032777  020000  175676          BIT     #20000,@C1      ;SEE IF ERROR
002612  001402                          BEQ     LT2B            ;IF NOT: BR
002614  004767  000024                  JSR     PC,LT2ER1       ;ELSE GO TO ERROR 1
002620  032777  000002  175676  LT2B:   BIT     #2,@ER          ;SEE IF ILR
002626  001402                          BEQ     LT2C            ;IF NOT: BR
002630  004767  000026                  JSR     PC,LT2ER2       ;ELSE GO TO ERROR 2
002634  022225                  LT2C:   CMP     (R2)+,(R5)+     ;BUMP ADDRESS
002636  005300                          DEC     R0
002640  001357                          BNE     LT2A            ;CONTINUE FOR ALL REGISTERS
002642  000435                          BR      LT2X
002644  012767  000002  176044  LT2ER1: MOV     #2,EXFL         ;FLAG NOT EXPECTED
002652  012767  025466  176012          MOV     #MSG4,ERADD     ;POINT TO CONTROLLER ERROR
002660  000415                          BR      LT2ERG          ;GO TO ERROR
002662  012767  000002  176026  LT2ER2: MOV     #2,EXFL         ;FLAG NOT EXPECTED
002670  012767  025504  175774          MOV     #MSG5,ERADD     ;POINT TO DRIVE ERROR
002676  000406                          BR      LT2ERG          ;GO TO ERROR
002700  012767  000001  176010  LT2ER3: MOV     #1,EXFL         ;FLAG EXPECTED
002706  012767  025466  175756          MOV     #MSG4,ERADD     ;POINT TO DRIVE
002714  012767  002730  175770  LT2ERG: MOV     #LT2LP,SCOLP    ;SET SCOPE ADDRESS
002722  004767  017066                  JSR     PC,LTGER        ;GO PRINT
002726  000207                          RTS     PC              ;ELSE CONTINUE
002730  005726                  LT2LP:  TST     (SP)+           ;RESET STACK
002732  000167  177642                  JMP     LT2A            ;LOOP
002736  004767  020456          LT2X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
002742  000167  177150                  JMP     TSCD2           ;RETURN TO SCHED
```

```
                                        ;LOGIC TEST 3: CONTROL BUS**********
                                LT3:    NOP
002746  000240                        
002750  012767  027543  175644  LT3IT:  MOV     #MSLT3,EMADDR   ;SET TEST HEADER
002756  012701  000001                  MOV     #1,R1           ;PRESET PATTERN 1
002762  012700  000020                  MOV     #20,R0          ;SET PATTERN CHANGE NUMBER
002766  004767  020510          LT3A:   JSR     PC,INIT1        ;GO INIT
002772  010177  175520                  MOV     R1,@#FC         ;WRITE TO FC
002776  032777  000010  175520          BIT     #10,@#ER        ;SEE IF CPAR (TM02)
003004  001013                          BNE     LT3ER1          ;IF SO: BR
003006  017702  175504          LT3B:   MOV     @#FC,R2         ;READ FC
003012  032777  020000  175470          BIT     #20000,@#C1     ;SEE IF NCPE (RH)
003020  001020                          BNE     LT3ER2          ;IF SO: BR
003022  005300                  LT3C:   DEC     R0              ;SEE IF DONE PATTERN CHANGES
003024  001427                          BEQ     LT3X            ;IF SO: BR
003026  000241                          CLC
003030  006101                          ROL     R1              ;CHANGE PATTERN
003032  000755                          BR      LT3A            ;CONTINUE
003034  012767  025775  175630  LT3ER1: MOV     #MSG11,ERADD    ;SET ERROR CODE
003042  012767  002766  175642          MOV     #LT3A,SCOLP     ;SET SCOPE ADDRESS
003050  017702  175442                  MOV     @#FC,R2         ;GET DATA
003054  004767  020074                  JSR     PC,LTGER1       ;GO DO ERROR
003060  000752                          BR      LT3B
003062  012767  025751  175602  LT3ER2: MOV     #MSG10,ERADD    ;SET ERROR CODE
003070  012767  003006  175614          MOV     #LT3B,SCOLP     ;SET SCOPE ADDRESS
003076  004767  020052                  JSR     PC,LTGER1       ;GO DO ERROR
003102  000747                          BR      LT3C
003104  105701                  LT3X:   TSTB    R1              ;SEE IF DONE PATTERN 2
003106  100405                          BMI     LT3XX           ;IF SO: BR
003110  012701  000401                  MOV     #401,R1         ;SET PATTERN 2
003114  012700  000010                  MOV     #10,R0          ;SET PATTERN CHANGE NUMBER
003120  000722                          BR      LT3A            ;DO PATTERN 2
003122  004767  020272          LT3XX:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
003126  000167  176764                  JMP     @#SCD2          ;RETURN TO SCHEDULAR
```

```
                                        ;LOGIC TEST 4: SLAVE ADDRESSING**********
003132  005767  175656          LT4:    TST     SKAT            ;SEE IF SKIP ADDRESS TESTS
003136  001403                          BEQ     LT4G0           ;IF NOT: BR
003140  005767  175574                  TST     STFLG           ;SEE IF SINGLE TEST
003144  001517                          BEQ     LT4X            ;IF NOT: BR
003146  012704  025600          LT4G0:  MOV     #MSG0A,R4
003152  004767  020666                  JSR     PC,TTOUT        ;PRINT TEST INSTRUCTIONS
003156  012704  025561          LT4G:   MOV     #MSG0,R4
003162  004767  020656                  JSR     PC,TTOUT        ;REQUEST SLAVE
003166  012705  000664                  MOV     #SLVN,R5
003172  012701  000001                  MOV     #1,R1
003176  012702  000007                  MOV     #7,R2
003202  012703  000008                  MOV     #8,R3
003206  004767  020374                  JSR     PC,TTR          ;GET SLAVE NUMBER
003212  005767  175456                  TST     TEMP1           ;SEE IF SLAVE
003216  001472                          BEQ     LT4X            ;IF NOT: BR
003220  005001                          CLR     R1              ;SELECT SLAVE 0
003222  012700  000010                  MOV     #10,R0          ;SET NUMBER OF SLAVES
003226  012777  000040  175264  LT4A:   MOV     #40,@CS         ;INIT
003234  016777  175364  175256          MOV     DRVN,@CS        ;SELECT DRIVE
003242  010177  175274                  MOV     R1,@TC          ;SELECT SLAVE
003246  017703  175264                  MOV     @DT,R3          ;GET DT
003252  020167  175406                  CMP     R1,SLVN         ;SEE IF SHOULD HAVE SPR
003256  001405                          BEQ     LT4B            ;IF SO: BR
003260  032703  002000                  BIT     #2000,R3        ;SEE IF SPR
003264  001417                          BEQ     LT4D            ;IF NOT: BR
003266  000167  000044                  JMP     LT4ER1          ;GO TO ERROR 1
003272  032703  002000          LT4B:   BIT     #2000,R3        ;SEE IF SPR
003276  001002                          BNE     LT4C            ;IF SO: BR
003300  000167  000042                  JMP     LT4ER2          ;ELSE GO TO ERROR
003304  012704  026466          LT4C:   MOV     #MSG30,R4
003310  004767  020530                  JSR     PC,TTOUT        ;PRINT SERIAL NUMBER TAG
003314  017703  175220                  MOV     @SN,R3
003320  004767  021534                  JSR     PC,SNPT         ;PRINT SERIAL NUMBER
003324  005300          LT4D:           DEC     R0
003326  001713                          BEQ     LT4G            ;IF DONE ALL: BR
003330  005201                          INC     R1              ;BUMP SLAVE
003332  000167  177670                  JMP     LT4A            ;CONTINUE
003336  012767  000001  175352  LT4ER1: MOV     #1,EXFL         ;FLAG EXPT: NOT RECEIVED
003344  000403                          BR      LT4ERG
003346  012767  000002  175342  LT4ER2: MOV     #2,EXFL         ;FLAG RECVD: NOT EXPT
003354  012767  027625  175240  LT4ERG: MOV     #MSLT4,EMADDR   ;SET  LT4 HEADER
003362  012767  025727  175302          MOV     #MSG0,ERADD     ;SET ERROR CONDITION
003370  012767  003226  175314          MOV     #LT4A,SCOLP     ;SET SCOPE ADDRESS
003376  004767  016412                  JSR     PC,LTGER        ;GO TO ERROR
003402  000750                          BR      LT4D            ;IF NO SCOPE: BR
003404  000167  176506          LT4X:   JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                ;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST**********
                                                
003410  012767  027704  175204  LT5:    MOV     #MSLT5,EMADDR   ;SET TEST HEADER
003416  004767  020060          LT5IT:  JSR     PC,INIT1        ;GO INIT
003422  012700  000032                  MOV     #32,R0          ;SET LOOP FOR BITS 4-0
003426  005001                          CLR     R1              ;SET TEST WORD
003430  010177  175100          LT5A:   MOV     R1,@MR          ;SEND TEST WORD TO MR
003434  017702  175074                  MOV     @MR,R2          ;READ MR
003440  042702  177740                  BIC     #177740,R2      ;MASK BITS 4-0
003444  020102                          CMP     R1,R2           ;SEE IF EXPT = RECVD
003446  001402                          BEQ     LT5B            ;IF SO: BR
003450  000167  000056                  JMP     LT5ER1          ;ELSE GO TO ERROR 1
003454  005300          LT5B:   DEC     R0
003456  001402                          BEQ     LT5C            ;IF DONE LOOP: BR
003460  005201                          INC     R1              ;BUMP TEST WORD
003462  000762                          BR      LT5A            ;CONTINUE LOOP
003464  012701  000015          LT5C:   MOV     #15,R1          ;SET TEST WORD + WAM 3
003470  012700  001000                  MOV     #1000,R0        ;SET LOOP FOR BITS 15-7
003474  010177  175034          LT5D:   MOV     R1,@MR          ;LOAD MR
003500  017702  175030                  MOV     @MR,R2          ;READ MR
003504  042702  000140                  BIC     #140,R2         ;MASK OUT BITS 5,6
003510  020102                          CMP     R1,R2           ;SEE IF EXPT = RECVD
003512  001402                          BEQ     LT5E            ;IF SO: BR
003514  000167  000036                  JMP     LT5ER2          ;ELSE GO TO ERR 2
003520  005300          LT5E:   DEC     R0
003522  001427                          BEQ     LT5X            ;IF DONE LOOP: BR
003524  062701  000200                  ADD     #200,R1         ;BUMP TEST WORD
003530  000761                          BR      LT5D            ;CONTINUE LOOP
003532  012767  026040  175132  LT5ER1: MOV     #MSG14,ERADD    ;SET ERROR CODE
003540  012767  003430  175144          MOV     #LT5A,SCOLP     ;SET SCOPE ADDRESS
003546  004767  017402                  JSR     PC,LTGER1       ;GO TO ERROR
003552  000167  177676                  JMP     LT5B            ;CONTINUE
003556  012767  026055  175106  LT5ER2: MOV     #MSG15,ERADD    ;SET ERROR CODE
003564  012767  003474  175120          MOV     #LT5D,SCOLP     ;SET SCOPE ADDRESS
003572  004767  017356                  JSR     PC,LTGER1       ;GO TO ERROR
003576  000167  177716                  JMP     LT5E            ;CONTINUE
003602  004767  017612          LT5X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
003606  000167  176304                  JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                ;LOGIC TEST 6: TC REGISTER BIT TEST**********
                                LT6:    NOP
003612  000240                  LT6IT:  MOV     #MSLT6,EMADDR   ;POINT TO LT6 HEADER
003614  012767  027750  175000          MOV     #3,R0           ;SET NUMBER OF TESTS
003622  012700  000003                  CLR     R1
003626  005001                  LT6A1:  JSR     PC,INIT1        ;GO INIT
003630  004767  017646          LT6A:   MOV     R1,@TC          ;WRITE TC
003634  010177  174702          LT6B:   MOV     @TC,R2          ;READ TC
003640  017702  174676                  BIC     #160000,R2      ;MASK OUT TCW
003644  042702  160000                  CMP     R1,R2           ;SEE IF EXPT = RECVD
003650  020102                          BEQ     LT6C            ;IF SO: BR
003652  001402                          JMP     LT6ER1          ;ELSE GO TO ERROR
003654  000167  000036                  BIT     #20000,@TC      ;SEE IF TCW SET
003660  032777  020000  174654  LT6C:   BNE     LT6D            ;IF SO: BR
003666  001002                          JMP     LT6ER2          ;ELSE GO TO ERROR
003670  000167  000046                  DEC     R0
003674  005300                  LT6D:   BEQ     LT6X            ;IF DONE ALL: BR
003676  001446                          CMP     #1,R0           ;SEE IF RESET TEST
003700  022700  000001                  BEQ     LT6A1           ;IF SO: BR
003704  001750                  MOV     #17777,R1       ;SET TEST WORD
003706  012701  017777                  JMP     LT6A            ;DO SET TEST
003712  000167  177712                  MOV     #MSG18,ERADD    ;SET ERROR CODE
003716  012767  026117  174746  LT6ER1: MOV     #LT6B,SCOLP     ;SET SCOPE ADDRESS
003724  012767  003634  174760          JSR     PC,LTGER1       ;GO TO ERROR
003732  004767  017216                  JMP     LT6C            ;CONTINUE
003736  000167  177716                  MOV     #MSG17,ERADD    ;SET ERROR CODE
003742  012767  026103  174722  LT6ER2: MOV     R1,SAV1         ;SAVE R1
003750  010167  174730                  MOV     #1,R1           ;SET EXPT = 1
003754  012701  000001                  CLR     R2              ;SET RECVD = 0
003760  005002                          MOV     #LT6ER4,SCOLP   ;SET SCOPE ADDRESS
003762  012767  004004  174722          JSR     PC,LTGER1       ;GO TO ERROR
003770  004767  017160                  MOV     SAV1,R1
003774  016701  174704                  JMP     LT6D            ;ELSE CONTINUE
004000  000167  177670          LT6ER4: CLR     @TC             ;WRITE TO TC
004004  005077  174532                  JMP     LT6C            ;LOOP ON ERROR
004010  000167  177644          LT6X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
004014  004767  017400                  JMP     TSCD2           ;RETURN TO SCHED
004020  000167  176072
```

```
                                                ;LOGIC TEST 7: FRAME COUNT BIT TEST**********
                                        LT7:    NOP
004024  000240                                  
004026  012700  000003                LT7IT:  MOV     #3,R0           ;SET TEST NUMBER
004032  012767  030014  174562        LT7C:   MOV     #MSLT7,EMADDR   ;SET TEST HEADER
004040  005001                                CLR     R1              ;SET TEST WORD
004042  004767  017434                LT7A:   JSR     PC,INIT1        ;GO INIT
004046  010177  174444                        MOV     R1,@#FC         ;CLEAR FRAME COUNT
004052  017702  174440                        MOV     @#FC,R2         ;READ FC
004056  020102                               CMP     R1,R2           ;SEE IF EXPT = RECVD
004060  001402                               BEQ     LT7B            ;IF SO: BR
004062  000167  000022                       JMP     LT7ER1          ;ELSE GO TO ERROR
004066  005300                        LT7B:   DEC     R0              ;SEE IF DONE ALL
004070  001421                               BEQ     LT7X            ;IF SO: BR
004072  022700  000001                       CMP     #1,R0           ;SEE IF RESET TEST
004076  001755                               BEQ     LT7C            ;IF SO: BR
004100  012701  177777                       MOV     #-1,R1          ;SET TEST WORD TO -1
004104  000167  177732                       JMP     LT7A            ;CONTINUE
004110  012767  026136  174554        LT7ER1: MOV     #MSG19,ERADD    ;SET ERROR CODE
004116  012767  004042  174566                MOV     #LT7A,SCOLP     ;SET SCOPE ADDRESS
004124  004767  017024                       JSR     PC,LTGER1       ;GO PRINT ERROR
004130  000167  177732                       JMP     LT7B            ;ELSE CONTINUE
004134  012700  000003                LT7X:   MOV     #3,R0           ;RESET TEST AMT
004140  004767  017254                       JSR     PC,ITER         ;GO SEE IF ITERATIONS
004144  000167  175746                       JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                ;LOGIC TEST 10: FUNCTION CODE BIT TEST**********

004150  000240                        LT10:   NOP
004152  012767  030060  174442       LT10IT: MOV     #MSLT10,EMADDR  ;SET TEST HEADER
004160  012700  000003               LT10A1: MOV     #3,R0           ;SET NUMBER OF TESTS
004164  005001                        LT10A:  CLR     R1              ;SET TEST WORRD
004166  012777  000040  174324       LT10A:  MOV     #40,@CS         ;INIT
004174  016777  174424  174316               MOV     DRVN,@CS        ;SELECT DRIVE
004202  010177  174302               LT10B:  MOV     R1,@C1          ;WRITE C1
004206  017702  174276               LT10E1: MOV     @C1,R2          ;READ C1
004212  042702  177701               LT10X:  BIC     #177701,R2      ;MASK FUNCTION CODE
004216  020102                       LT10E:  CMP     R1,R2           ;SEE IF EXPT = RECVD
004220  001402                               BEQ     LT10B           ;IF SO: BR
004222  000167  000022                       JMP     LT10E1          ;ELSE GO TO ERROR
004226  005300                        LT10B:  DEC     R0
004230  001421                               BEQ     LT10X           ;IF DONE ALL: BR
004232  022700  000001                       CMP     #1,R0           ;SEE IF RESET TEST
004236  001752                               BEQ     LT10A1          ;IF SO: BR
004240  012701  000076                       MOV     #76,R1          ;SET TEST WORD
004244  000167  177716                       JMP     LT10A           ;DO SET TEST
004250  012767  026155  174414       LT10E1: MOV     #MSG20,ERADD    ;SET ERROR CODE
004256  012767  004166  174426               MOV     #LT10A,SCOLP    ;SET SCOPE ADDRESS
004264  004767  016664                       JSR     PC,LTGER1       ;GO PRINT ERROR
004270  000167  177732                       JMP     LT10B           ;ELSE CONTINUE
004274  004767  017120                LT10X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
004300  000167  175612                       JMP     TSCD2           ;RETURN TO SCHED
```

```
                                        ;LOGIC TEST 11: GO BIT SET RESET**********

004304  000240                LT11:   NOP
004306  012767  030133  174306  LT11IT: MOV     #MSLT11,EMADDR  ;SET TEST HEADER
004314  004767  017162          JSR     PC,INIT1        ;GO INIT
004320  017702  174164          MOV     @C1,R2          ;READ C1
004324  032702  000001          BIT     #1,R2           ;SEE IF GO=0
004330  001402                  BEQ     LT11B           ;IF SO: BR
004332  000167  000066          JMP     LT11E1          ;ELSE GO TO ERROR 1
004336  012777  000015  174170  LT11B:  MOV     #15,@MR         ;SELECT MAM 3
004344  005077  174146          CLR     @FC             ;ASSURE FCS = 1
004350  052777  001700  174164          BIS     #1700,@TC       ;ASSURE FMT OK
004356  012777  000071  174124          MOV     #71,@C1         ;SET READ+GO
004364  017702  174120          MOV     @C1,R2          ;READ C1
004370  032702  000001          BIT     #1,R2           ;SEE IF GO =1
004374  001002                  BNE     LT11C           ;IF SO: BR
004376  000167  000054          JMP     LT11E2          ;ELSE GO TO ERROR 2
004402  004767  017074          LT11C:  JSR     PC,INIT1        ;GO INIT
004406  017702  174076          MOV     @C1,R2          ;READ C1
004412  032702  000001          BIT     #1,R2           ;SEE IF GO=0
004416  001447                  BEQ     LT11X           ;IF SO;BR
004420  000167  000064          JMP     LT11E3          ;ELSE GO TO ERROR 3
004424  012767  026207  174240  LT11E1: MOV     #MSG21,ERADD    ;SET ERROR CODE
004432  012702  000001          MOV     #1,R2           ;SET RCVD
004436  005001                  CLR     R1              ;SET EXPT
004440  012767  004306  174244          MOV     #LT11IT,SCOLP           ;SET SCOPE ADDRESS
004446  004767  016502          JSR     PC,LTGER1       ;GO PRINT ERROR
004452  000167  177660          JMP     LT11B           ;ELSE CONTINUE
004456  012767  026245  174206  LT11E2: MOV     #MSG22,ERADD    ;SET ERROR CODE
004464  005002                  CLR     R2              ;SET RCVD
004466  012701  000001          MOV     #1,R1           ;SET EXPT
004472  012767  004336  174212          MOV     #LT11B,SCOLP    ;SET SCOPE ADDRESS
004500  004767  016450          JSR     PC,LTGER1       ;GO PRINT ERROR
004504  000167  177672          JMP     LT11C           ;ELSE CONTINUE
004510  012767  026266  174154  LT11E3: MOV     #MSG23,ERADD    ;SET ERROR CODE
004516  005001                  CLR     R1              ;SET EXPT
004520  012702  000001          MOV     #1,R2           ;SET RCVD
004524  012767  004402  174160          MOV     #LT11C,SCOLP    ;SET SCOPE ADDRESS
004532  004767  016416          JSR     PC,LTGER1       ;GO PRINT ERROR
004536  004767  016656          LT11X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
004542  000167  175350          JMP     TSCD2           ;RETURN TO SCHED
```

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

004546  000240                LT12:   NOP
004550  012767  030200  174044 LT12IT: MOV     #MSLT12,EMADDR  ;SET TEST HEADER
004556  004767  016720              JSR     PC,INIT1        ;GO INIT
004562  032777  000200  173732        BIT     #200,@DS        ;SEE IF DRY=1
004570  001002                      BNE     LT12B           ;IF SO: BR
004572  000167  000062              JMP     LT12E1          ;ELSE GO TO ERROR 1
004576  012777  000015  173730 LT12B: MOV     #15,@MR         ;SET WAM3
004604  005077  173706              CLR     @FC             ;ASSURE FCS = 1
004610  052777  001700  173724        BIS     #1700,@TC       ;ASSURE FMT OK
004616  012777  000071  173664        MOV     #71,@C1         ;SET READ+GO
004624  032777  000200  173670        BIT     #200,@DS        ;SEE IF DRY=0
004632  001402                      BEQ     LT12C           ;IF SO: BR
004634  000167  000042              JMP     LT12E2          ;ELSE GO TO ERROR 2
004640  004767  016636        LT12C: JSR     PC,INIT1        ;GO INIT
004644  032777  000200  173650        BIT     #200,@DS        ;SEE IF DRY=1
004652  001034                      BNE     LT12X           ;IF SO: BR
004654  000167  000044              JMP     LT12E3          ;ELSE GO TO ERROR 3
004660  012767  026321  174004 LT12E1: MOV     #MSG24,ERADD    ;SET ERROR CODE
004666  012767  004550  174016        MOV     #LT12IT,SCOLP   ;SET SCOPE ADDRESS
004674  004767  016246              JSR     PC,LTGER2       ;GO TO ERROR
004700  000736                      BR      LT12B           ;CONTINUE
004702  012767  026347  173762 LT12E2: MOV     #MSG25,ERADD    ;SET ERROR CODE
004710  012767  004576  173774        MOV     #LT12B,SCOLP    ;SET LOOP ADDRESS
004716  004767  016224              JSR     PC,LTGER2       ;GO PRINT ERROR
004722  000746                      BR      LT12C           ;CONTINUE
004724  012767  026376  173740 LT12E3: MOV     #MSG25A,ERADD   ;SET ERROR CODE
004732  012767  004640  173752        MOV     #LT12C,SCOLP    ;SET ERROR LOOP
004740  004767  016202              JSR     PC,LTGER2       ;GET PRINT ERROR
004744  004767  016450        LT12X: JSR     PC,ITER         ;GO TO ITERATION SUBROUTINE
004750  000167  175142              JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                ;LOGIC TEST 13: INTERRUPT TEST**********

004754  005000                        LT13:   CLR     R0
004756  012767  030251  173636                MOV     #MSLT13,EMADDR  ;SET TEST HEADER
004764  004767  016512                LT13IT: JSR     PC,INIT1        ;GO INIT,SELECT DRIVE, SELECT ABOVE
004770  012767  005046  173672                MOV     #LT13X,RTRN     ;SET RETURN ADDRESS
004776  005077  173506                        CLR     @C1             ;CLEAR CS1
005002  005077  173560                        CLR     @PSW            ;SET PRIORITY
005006  052777  000100  173474                BIS     #100,@C1        ;BIT SET IE
005014  005300                        LT13A:  DEC     R0
005016  001376                                BNE     LT13A           ;AWAIT INTERRUPT
005020  012777  000340  173540        LT13E1: MOV     #340,@PSW       ;RESET PRIORITY
005026  012767  026423  173636                MOV     #MSG26,ERADD    ;SET ERROR CODE
005034  012767  004764  173650                MOV     #LT13IT,SCOLP   ;SET LOOP ADDRESS
005042  004767  016100                        JSR     PC,LTGER2       ;GO PRINT ERROR
005046  004767  016346                LT13X:  JSR     PC,ITER         ;GO TO ITERATION SUBROUTINE
005052  000167  175040                        JMP     TSCD2           ;RETURN TO SCHED
```

```
                                        ;THE NEXT 4 TESTS ARE MANUAL INTERVENTION STATUS TESTS,
                                        ;THE OPERATOR WILL BE REQUIRED TO MANIPULATE THE TU16
                                        ;CONTROL PANEL IN ACCORDANCE WITH TTY INSTRUCTIONS.

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

005056  032777  001000  173504  LT14:   BIT     #1000,@#WR      ;SEE IF INHIB MAN TST
005064  001005                          BNE     LT14A           ;IF NOT: BR
005066  005767  173646                  TST     $TFLG           ;SEE IF SINGLE TEST
005072  001433                          BEQ     LT14XX          ;IF NOT: BR
005074  000167  016362                  JMP     INHT            ;ELSE GO PRINT INHIB MSG
005100  012767  030316  173514  LT14A:  MOV     #MSLT14,EMADDR  ;SET TEST HEADER
005106  012704  033237                  MOV     #MMSG1,R4       ;SET INSTRUCTION ONE
005112  004767  016410                  JSR     PC,INST         ;GO DO INSTRUCTION
005116  004767  016360          LT14IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE + SLAVE
005122  012701  014602                  MOV     #14602,R1       ;SET TEST WORD
005126  017702  173370                  MOV     @DS,R2          ;ASSURE MOL,WRL,DPR,DRY,BOT
005132  020102                          CMP     R1,R2
005134  001410                          BEQ     LT14X           ;IF SO: BR
005136  012767  005116  173546          MOV     #LT14IT,$COLP   ;SET LOOP ADDRESS
005144  012767  026452  173520          MOV     #MSG27,ERADD    ;SET ERROR CODE
005152  004767  015776                  JSR     PC,LTGER1       ;GO PRINT ERROR
005156  004767  016236          LT14X:  JSR     PC,ITER         ;GO SEE IF ITERATION
005162  000167  174738          LT14XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
                                        ;LOGIC TEST 15: STATUS AT BOT, OFFLINE, LOADED, NO WRITE RING**********
 005166  032777  001000  173374  LT15:   BIT     #1000,@#WR      ;SEE IF INHIB MAN TST
 005174  001005                         BNE     LT15A           ;IF NOT: BR
 005176  005767  173536                 TST     $TFLG           ;SEE IF SINGLE TEST
 005202  001433                         BEQ     LT15XX          ;IF NOT: BR
 005204  000167  016252                 JMP     INMT            ;ELSE GO PRINT INHIB MSG
 005210  012767  030364  173404  LT15A: MOV     #MSLT15,EMADDR  ;SET TEST HEADER
 005216  012704  033335                 MOV     #MMSG2,R4
 005222  004767  016300                 JSR     PC,INST         ;PRINT INSTRUCTION
 005226  004767  016256         LT15IT: JSR     PC,INIT2        ;GO INIT, SELECT DRIVE, SLAV
 005232  012701  100700                 MOV     #100700,R1      ;SET TEST WORD
 005236  017702  173260                 MOV     @DS,R2          ;READ STATUS
 005242  020102                         CMP     R1,R2           ;SEE OF EXPT=RCVD
 005244  001410                         BEQ     LT15X
 005246  012767  005226  173436         MOV     #LT15IT,$COLP   ;SET LOOP ADDRESS
 005254  012767  026452  173410         MOV     #MSG27,ERADD    ;SET ERROR CODE
 005262  004767  015666                 JSR     PC,LTGER1       ;GO PRINT ERROR
 005266  004767  016126         LT15X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
 005272  000167  174620         LT15XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
                                        ;LOGIC TEST 16: STATUS AT EOT, OFFLINE LOADED, NO WRITE RING**********
 005276  032777  001000  173264  LT16:  BIT     #1000,@#WR       ;SEE IF INHIB MAN TST
 005304  001005                       BNE     LT16A            ;IF NOT: BR
 005306  005767  173426               TST     @TFLG            ;SEE IF SINGLE TEST
 005312  001433                       BEQ     LT16XX           ;IF NOT: BR
 005314  000167  016142               JMP     INMT             ;ELSE GO PRINT INHIB MSG
 005320  012767  030432  173274  LT16A: MOV     #MSLT16,EMADDR   ;SET TEST HEADER
 005326  012704  033356               MOV     #MMSG3,R4
 005332  004767  016170               JSR     PC,INST          ;GO PRINT INSTRUCTION
 005336  004767  016146        LT16IT: JSR     PC,INIT2         ;SELECT DRIVE,SLAVE
 005342  012701  116701               MOV     #116701,R1       ;SET TEST WORD
 005346  017702  173150               MOV     @DS,R2           ;READ STATUS
 005352  020102                       CMP     R1,R2            ;SEE IF EXPT=RCVD
 005354  001410                       BEQ     LT16X            ;IF SO: BR
 005356  012767  005336  173326       MOV     #LT16IT,@COLP    ;SET LOOP ADDRESS
 005364  012767  026452  173300       MOV     #MSG27,ERADD     ;SET ERROR CODE
 005372  004767  015556               JSR     PC,LTGER1        ;GO PRINT ERROR
 005376  004767  016016        LT16X: JSR     PC,ITER          ;GO SEE IF ITERATION
 005402  000167  174510        LT16XX: JMP    TSCD2            ;RETURN TO SCHED
```

```
                                        ;LOGIC TEST 17: STATUS AT ON LINE, LOADED**********
                                        ;
005406  032777  001000  173154  LT17:   BIT     #1000,@#WR      ;SEE IF INHIB MAN TST
005414  001005                          BNE     LT17A           ;IF NOT: BR
005416  005767  173316                  TST     STFLG           ;SEE IF SINGLE TEST
005422  001433                          BEQ     LT17XX          ;IF NOT: BR
005424  000167  016032                  JMP     INHT            ;ELSE GO PRINT INHIB MSG
005430  012767  030500  173164  LT17A:  MOV     #MSLT17,EMADDR  ;SET TEST HEADER
005436  012704  033414                  MOV     #MMSG4,R4
005442  004767  016060                  JSR     PC,INST         ;GO PRINT INSTRUCTION
005446  004767  016036          LT17IT: JSR     PC,INIT2        ;SELECT DRIVE, SLAVE
005452  012701  110701                  MOV     #110701,R1      ;SET TEST WORD
005456  017702  173040                  MOV     @DS,R2          ;READ STATUS
005462  020102                          CMP     R1,R2           ;SEE IF EXPT=RCVD
005464  001410                          BEQ     LT17X           ;IF SO: BR
005466  012767  005446  173216          MOV     #LT17IT,SCOLP   ;SET LOOP ADDRESS
005474  012767  026452  173170          MOV     #MSG27,ERADD    ;SET ERROR CODE
005502  004767  015446                  JSR     PC,LTGER1       ;YES PRINT ERROR
005506  004767  015706          LT17X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
005512  000167  174400          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 TM02
                                        ;FOR EACH ERROR CONDITION SET THE APPROPRIATE STATUS WILL BE
                                        ;CHECKED, IE; ERR, ATA, SLA, SC ETC,

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

005516  012767  030546  173076  LT20:   MOV     #MSLT20,EMADDR  ;SET TEST HEADER
005524  012767  005544  173160          MOV     #LT20A,SCOLP    ;SET LOOP ADDRESS
005532  012700  000022          LT20IT: MOV     #22,R0          ;SET NUMBER OF ILL CODES
005536  012767  000544  173130          MOV     #ILFT,TEMP1     ;POINT TO START IF TABLE
005544  004767  015732          LT20A:  JSR     PC,INIT1        ;GO INIT, SELECT SLAVE + DRIVE
005550  012777  177777  172734          MOV     #-1,@WC         ;SET WC= -1
005556  012701  000001                  MOV     #1,R1           ;SET TEST WORD
005562  117777  173106  172720          MOVB    @TEMP1,@C1      ;SET ILL CODE
005570  017702  172730                  MOV     @ER,R2          ;READ ER
005574  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
005576  001011                          BNE     LT20B           ;IF SO; BR
005600  012767  033677  173064          MOV     #TM$17,ERADD    ;SET ERROR CODE
005606  012767  000001  173102          MOV     #1,EXFL         ;SET EXPT FLG
005614  004767  014166                  JSR     PC,LTGER@       ;GO PRINT ERROR
005620  000404                          BR      LT20C
005622  020102          LT20B:  CMP     R1,R2           ;SEE UNEXPECTED ERRORS
005624  001402                  BEQ     LT20C           ;IF NOT; BR
005626  004767  014142                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
005632  005300          LT20C:  DEC     R0              ;SEE IF DONE ALL ILL CODES
005634  001404                  BEQ     LT20X           ;IF SO; BR
005636  005267  173032                  INC     TEMP1           ;BUMP ADDRESS
005642  000167  177676                  JMP     LT20A           ;CONTINUE
005646  004767  015546          LT20X:  JSR     PC,ITER         ;GO SEE IF ITERATION
005652  004767  014606                  JSR     PC,DRVCLR
005656  000167  174234                  JMP     TSCD2           ;RETURN TO SCHED
```

```
1010
1011                                        ;LOGIC TEST 21: REGISTER MODIFICATION REFUSED(RMR)**********
1012
1013  005662  012767  030625  172732  LT21:   MOV     #MSLT21,EMADDR  ;SET TEST HEADER
1014  005670  012767  005676  173014          MOV     #LT21IT,SCOLP   ;SET SCOPE LOOP ADDRESS
1015  005676  004767  015600          LT21IT: JSR     PC,INIT1        ;GO INIT, SELECT SLAVE, DRIVE
1016  005702  052777  000300  172632          BIS     #300,@TC        ;SET FORMAT
1017  005710  012777  000015  172616          MOV     #15,@MR         ;SET WAM3
1018  005716  012777  000071  172564          MOV     #71,@C1         ;SET READ+GO
1019  005724  005077  172566                  CLR     @FC             ;ATTEMPT WRITE TO FC
1020  005730  012701  000004                  MOV     #4,R1           ;SET TEST WORD
1021  005734  017702  172564                  MOV     @ER,R2          ;GET ER
1022  005740  030102                          BIT     R1,R2           ;SEE IF EXPT+RCVD
1023  005742  001011                          BNE     LT21A           ;IF SO: BR
1024  005744  012767  033713  172720          MOV     #TM819,ERADD    ;SET ERROR CODE
1025  005752  012767  000001  172736          MOV     #1,EXFL         ;SET EXPT FLG
1026  005760  004767  014022                  JSR     PC,LTGER0       ;GO PRINT ERROR
1027  005764  000404                          BR      LT21B
1028  005766  020102          LT21A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
1029  005770  001402                          BEQ     LT21B           ;IF NOT: BR
1030  005772  004767  013776                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
1031  005776  004767  015416          LT21B:  JSR     PC,ITER         ;GO SEE IF ITERATION
1032  006002  012703  040000                  MOV     #40000,R3
1033  006006  005303          LT21XA: DEC     R3              ;DELAY FOR ALPHA
1034  006010  001376                          BNE     LT21XA
1035  006012  004767  013560                  JSR     PC,EORPA        ;GO DO EOR CLEAR
1036  006016  004767  014442                  JSR     PC,DRVCLR
1037  006022  000167  174070                  JMP     TSCD2           ;RETURN TO SCHED
```

```
1038
1039                                        ;LOGIC TEST 22: CONTROL BUS PARITY (CPAR)**********
1040
1041  006026  012767  030661  172566  LT22:   MOV     #MSLT22,EMADDR  ;SET TEST HEADER
1042  006034  012767  006042  172650          MOV     #LT22IT,SCOLP   ;SET SCOPE LOOP ADDRESS
1043  006042  004767  015434          LT22IT: JSR     PC,INIT1        ;INIT, SELECT SLAVE+DRIVE
1044  006046  052777  000020  172444          BIS     #20,@CS         ;ENABLE EVEN PARITY ON MB
1045  006054  012777  177777  172434          MOV     #-1,@FC         ;WRITE TO FC
1046  006062  012701  000010                  MOV     #10,R1          ;SET TEST WORD
1047  006066  042777  000020  172424          BIC     #20,@CS         ;RESET PARITY TO ODD
1048  006074  017702  172424                  MOV     @ER,R2          ;GET ER
1049  006100  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
1050  006102  001011                          BNE     LT22A           ;IF SO: BR
1051  006104  012767  033721  172560          MOV     #TMS20,ERADD    ;SET ERROR CODE
1052  006112  012767  000001  172576          MOV     #1,EXFL         ;SET EXPT FLG
1053  006120  004767  013662                  JSR     PC,LTGER0       ;GO PRINT ERROR
1054  006124  000404                          BR      LT22X
1055  006126  020102          LT22A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
1056  006130  001402                          BEQ     LT22X           ;IF NOT: BR
1057  006132  004767  013636                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
1058  006136  004767  015256          LT22X:  JSR     PC,ITER         ;GO SEE IF ITERATION
1059  006142  004767  014316                  JSR     PC,DRVCLR
1060  006146  000167  173744                  JMP     TSCD2           ;RETURN TO SCHED
```

```
1061
1062                                        ;LOGIC TEST 23: FORMAT ERROR(FMT)**********
1063
1064  006152  012767  030716  172442  LT23:   MOV     #MSLT23,EMADDR  ;SET TEST HEADER
1065  006160  012767  006166  172524          MOV     #LT23IT,SCOLP   ;SET SCOPE ADDRESS
1066  006166  004767  015310          LT23IT: JSR     PC,INIT1        ;GO INIT SELECT DRIVE+SLAVE
1067  006172  042777  000360  172342          BIC     #360,@TC        ;SET ILLEGAL FORMAT
1068  006200  012701  000020                  MOV     #20,R1          ;SET TEST WORD
1069  006204  012777  000015  172322          MOV     #15,@MR         ;SET WAM 3
1070  006212  012777  000071  172270          MOV     #71,@C1         ;SET READ+GO
1071  006220  017702  172300                  MOV     @ER,R2          ;READ ER
1072  006224  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
1073  006226  001011                          BNE     LT23A           ;IF SO: BR
1074  006230  012767  033730  172434          MOV     #TM821,ERADD    ;SET ERROR CODE
1075  006236  012767  000001  172452          MOV     #1,EXFL         ;SET EXPT FLG
1076  006244  004767  013536                  JSR     PC,LTGER@       ;GO PRINT ERROR
1077  006250  000404                          BR      LT23X
1078  006252  020102          LT23A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
1079  006254  001402                          BEQ     LT23X           ;IF NOT: BR
1080  006256  004767  013512                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
1081  006262  004767  015132          LT23X:  JSR     PC,ITER         ;GO SEE IF ITERATION
1082  006266  004767  013304                  JSR     PC,EORPA
1083  006272  004767  014166                  JSR     PC,DRVCLR
1084  006276  000167  173614                  JMP     TSCD2           ;RETURN TO SCHED
```

```
1085                                        ;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)**********
1086
1087  006302  012767  030760  172312  LT24:   MOV     #MSLT24,EMADDR  ;SET TEST HEADER
1088  006310  012767  006316  172374         MOV     #LT24IT,$COLP   ;SET SCOPE ADDRESS
1089  006316  012767  000005  172262  LT24IT: MOV     #5,ITAMT
1090  006324  004767  015152                 JSR     PC,INIT1        ;GO INIT, SELECT DRIVE+SLAVE
1091  006330  052777  000300  172204         BIS     #300,@TC        ;SET NORMAL FORMAT
1092  006336  012777  034144  172150         MOV     #WDATA,@BA      ;SET BA
1093  006344  012777  177760  172144         MOV     #-20,@FC        ;SET FC
1094  006352  012777  177770  172132         MOV     #-10,@WC        ;SET WC
1095  006360  012777  000013  172146         MOV     #13,@MR         ;SELECT WAM 2
1096  006366  012777  000061  172114         MOV     #61,@C1         ;SET WRITE+GO
1097  006374  052777  000020  172116         BIS     #20,@CS         ;FORCE EVEN PARITY
1098  006402  012701  000040                 MOV     #40,R1          ;SET TEST WORD
1099  006406  012703  000004                 MOV     #4,R3
1100  006412  005000                         CLR     R0
1101  006414  005300                 LT24A:  DEC     R0
1102  006416  001376                         BNE     LT24A           ;DELAY
1103  006420  005303                         DEC     R3
1104  006422  001374                         BNE     LT24A
1105  006424  012700  000004                 MOV     #4,R0
1106  006430  012777  000013  172076  LT24B:  MOV     #13,@MR         ;CLOCK MR 4 TIMES
1107  006436  005300                         DEC     R0
1108  006440  022700  000002                 CMP     #2,R0           ;SEE IF DONE 1 BYTE
1109  006444  001002                         BNE     LT24B0          ;IF NOT: BR
1110  006446  017701  172062                 MOV     @MR,R1          ;ELSE GET BYTE 1
1111  006452  005700                 LT24B0: TST     R0              ;SEE IF BYTE 2
1112  006454  001365                         BNE     LT24B           ;IF NOT: BR
1113  006456  017704  172052                 MOV     @MR,R4          ;GET BYTE 2
1114  006462  005000                         CLR     R0
1115  006464  005300                 LT24C:  DEC     R0
1116  006466  001376                         BNE     LT24C           ;DELAY
1117  006470  032777  000040  172026         BIT     #40,@ER         ;SEE IF DPAR IS SET
1118  006476  001023                         BNE     LT24D           ;IF SO: BR
1119  006500  000301                         SWAB    R1
1120  006502  042701  177400                 BIC     #177400,R1      ;GET LOW BYTE
1121  006506  042704  000377                 BIC     #377,R4
1122  006512  050401                         BIS     R4,R1           ;GET HIGH BYTE
1123  006514  005267  172224                 INC     T24FL           ;SET T24 FLAG
1124  006520  012767  033736  172144         MOV     #TM822,ERADD    ;SET ERROR CODE
1125  006526  012767  000001  172162         MOV     #1,EXFL         ;SET EXPT FLG
1126  006534  004767  013246                 JSR     PC,LTGER0       ;GO PRINT ERROR
1127  006540  005067  172200                 CLR     T24FL           ;CLEAR FLAG
1128  006544  000412                         BR      LT24X
1129  006546  012701  000050          LT24D:  MOV     #50,R1
1130  006552  017702  171746                 MOV     @ER,R2          ;GET ERROR REGISTER
1131  006556  042702  020000                 BIC     #20000,R2       ;MASK OPI
1132  006562  020102                         CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
1133  006564  001402                         BEQ     LT24X           ;IF NOT: BR
1134  006566  004767  013202                 JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
1135  006572  042777  000020  171720  LT24X:  BIC     #20,@CS         ;RESET EVEN PARITY
1136  006600  004767  012772                 JSR     PC,EORPA        ;GO DO EOR CLEAR
1137  006604  004767  013654                 JSR     PC,DRVCLR       ;GO SEE IF DRIVE CLEAR OK
1138  006610  004767  014604                 JSR     PC,ITER         ;GO SEE IF ITERATION
1139  006614  012767  000020  171764         MOV     #20,ITAMT
1140  006622  000167  173270                 JMP     [illegible]     ;RET[illegible]
```

```
 1141
 1142                                        ;LOGIC TEST 25: NON-EXECUTABLE FUNCTION(NEF)**********
 1143
 1144  006626  012767  031020  171766  LT25:   MOV     #MSLT25,EMADDR  ;SET TEST HEADER
 1145  006634  012767  006642  172050          MOV     #LT25IT,SCOLP   ;SET LOOP ADDRESS
 1146  006642  004767  014634        LT25IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE+SLAVE
 1147  006646  052777  000300  171666          BIS     #300,@TC        ;SET NORMAL FORMAT
 1148  006654  012777  177777  171634          MOV     #-1,@FC         ;SET ILLEGAL FC
 1149  006662  012777  000013  171644          MOV     #13,@MR         ;SET WAM 2
 1150  006670  012777  000061  171612          MOV     #61,@C1         ;LOAD WRITE+GO
 1151  006676  012701  004000                  MOV     #4000,R1        ;SET TEST WORD
 1152  006702  017702  171616                  MOV     @ER,R2          ;GET ER
 1153  006706  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
 1154  006710  001011                          BNE     LT25A           ;IF SO: BR
 1155  006712  012767  034024  171752          MOV     #TM831,ERADD    ;SET ERROR CODE
 1156  006720  012767  000001  171770          MOV     #1,EXFL         ;SET EXPT FLAG
 1157  006726  004767  013054                  JSR     PC,LTGER0       ;GO PRINT ERROR
 1158  006732  000404                          BR      LT25X
 1159  006734  020102                LT25A:    CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
 1160  006736  001402                          BEQ     LT25X           ;IF NOT: BR
 1161  006740  004767  013030                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
 1162  006744  004767  014450        LT25X:    JSR     PC,ITER         ;GO SEE IF ITERATION
 1163  006750  004767  013510                  JSR     PC,DRVCLR
 1164  006754  000167  173136                  JMP     TSCD2           ;RETURN TO SCHED
```

```
 1165
 1166                                           ;LOGIC TEST 26: FRAME COUNT ERROR(FCE)**********
 1167
 1168  006760  012767  031054  171634  LT26:   MOV     #MSLT26,EMADDR  ;SET TEST HEADER
 1169  006766  004767  014510          LT26IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE+SLAVE
 1170  006772  005000                          CLR     R0
 1171  006774  005300                  LT26W:  DEC     R0
 1172  006776  001376                          BNE     LT26W           ;AWAIT OPI RESET
 1173  007000  052777  000300  171534          BIS     #300,@TC        ;SET NORMAL FORMAT
 1174  007006  012777  177770  171476          MOV     #-10,@WC        ;SET WC=-10
 1175  007014  012777  177760  171474          MOV     #-20,@FC        ;SET FC=-20
 1176  007022  012777  000013  171504          MOV     #13,@MR         ;SET WAM 3
 1177  007030  012777  000061  171452          MOV     #61,@C1         ;LOAD WRITE+GO
 1178  007036  012701  001000                  MOV     #1000,R1        ;SET TEST WORD
 1179  007042  005000                          CLR     R0
 1180  007044  005300                  LT26A:  DEC     R0
 1181  007046  001376                          BNE     LT26A           ;DELAY
 1182  007050  012777  000025  171456          MOV     #25,@MR         ;LOAD MM EOR CLEAR
 1183  007056  105077  171452                  CLRB    @MR             ;RESET MR
 1184  007062  012703  000004                  MOV     #4,R3
 1185  007066  005000                          CLR     R0
 1186  007070  032777  001000  171426  LT26B:  BIT     #1000,@ER       ;SEE IF FCE SET
 1187  007076  001022                          BNE     LT26C           ;IF SO: BR
 1188  007100  005300                          DEC     R0
 1189  007102  001372                          BNE     LT26B           ;DELAY
 1190  007104  005303                          DEC     R3
 1191  007106  001370                          BNE     LT26B
 1192  007110  017702  171410                  MOV     @ER,R2          ;GET ER
 1193  007114  012767  006766  171570          MOV     #LT26IT,$COLP   ;SET SCOPE ADDRESS
 1194  007122  012767  034003  171542          MOV     #TM820,ERADD
 1195  007130  012767  000001  171560          MOV     #1,EXFL         ;SET EXPT FLG
 1196  007136  004767  012644                  JSR     PC,LTGER0       ;GO PRINT ERROR
 1197  007142  000406                          BR      LT26X
 1198  007144  017702  171354          LT26C:  MOV     @ER,R2          ;GET ERROR REGISTER
 1199  007150  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
 1200  007152  001402                          BEQ     LT26X           ;IF NOT: BR
 1201  007154  004767  012614                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
 1202  007160  004767  014234          LT26X:  JSR     PC,ITER         ;GO SEE IF ITERATION
 1203  007164  004767  013274                  JSR     PC,DRVCLR
 1204  007170  000167  172722                  JMP     TSCD2           ;RETURN TO SCHED
```

```
1205
1206                                                ;LOGIC TEST 27: ILLEGAL REGISTER(ILR)**********
1207
1208  007174  022767  172400  171306  LT27:   CMP     #172400,C1      ;SEE IF ADDRESSES OPEN
1209  007202  001041                          BNE     LT27XX          ;IF NOT: BR
1210  007204  012767  007230  171500          MOV     #LT27A,SCOLP    ;SET SCOPE ADDRESS
1211  007212  012767  031110  171402          MOV     #MSLT27,EMADDR  ;SET TEST HEADER
1212  007220  012700  000020          LT27IT: MOV     #20,R0          ;SET NUMBER OF ILR TESTS
1213  007224  012701  172434                  MOV     #172434,R1      ;SET FIRST ILR ADDRESS
1214  007230  004767  014246          LT27A:  JSR     PC,INIT1        ;GO INIT, SELECT DRIVE+SLAVE
1215  007234  011103                          MOV     (R1),R3         ;ATTEMPT ILR READ
1216  007236  032777  000002  171260          BIT     #2,@ER          ;SEE IF ILR=1
1217  007244  001010                          BNE     LT27B           ;IF SO: BR
1218  007246  012767  000001  171442          MOV     #1,EXPL         ;SET EXPT-NOT RCVD FLAG
1219  007254  012767  033625  171410          MOV     #TM810,ERADD    ;SET ERROR CODE
1220  007262  004767  012526                  JSR     PC,LTGER        ;GO PRINT ERROR
1221  007266  005300                  LT27B:  DEC     R0              ;SEE IF DONE ALL
1222  007270  001402                          BEQ     LT27X           ;IF SO: BR
1223  007272  005721                          TST     (R1)+           ;BUMP ADDRESS
1224  007274  000755                          BR      LT27A           ;CONTINUE TESTS
1225  007276  004767  014116          LT27X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
1226  007302  004767  013156                  JSR     PC,DRVCLR
1227  007306  000167  172604          LT27XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
1228
1229                                        ;LOGIC TEST 30: DRIVE TIMING ERROR**********
1230
1231  007312  012767  034032  171352  LT30:   MOV     #TM832,ERADD    ;SET ERROR CODE
1232  007320  012767  031144  171274          MOV     #MSLT30,EMADDR  ;SET TEST HEADER
1233  007326  012767  007334  171356          MOV     #LT30IT,SCOLP   ;SET SCOPE ADDRESS
1234  007334  004767  014142          LT30IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE + SLAVE
1235  007340  052777  000300  171174          BIS     #300,@TC        ;SET NORMAL FORMAT
1236  007346  012701  010000                  MOV     #10000,R1       ;SET TEST WORD
1237  007352  012777  000017  171154          MOV     #17,@MR         ;CRIPPLE OCCUPIED
1238  007360  005077  171132                  CLR     @FC             ;SET FC3
1239  007364  012777  000061  171116          MOV     #61,@C1         ;LOAD WRITE+GO
1240  007372  032777  010000  171124          BIT     #10000,@ER      ;SEE IF DTE SET
1241  007400  001005                          BNE     LT30A           ;IF SO: BR
1242  007402  012767  000001  171306          MOV     #1,EXFL         ;SET EXPT FLG
1243  007410  004767  012372                  JSR     PC,LTGER0       ;GO PRINT ERROR
1244  007414  004767  014062          LT30A:  JSR     PC,INIT1        ;GO INIT SELECT DRIVE,SLAVE
1245  007420  052777  000300  171114          BIS     #300,@TC        ;SET FORMAT
1246  007426  012701  010000                  MOV     #10000,R1       ;SET TEST WORD
1247  007432  005077  171060                  CLR     @FC             ;SET FC0
1248  007436  012777  000015  171070          MOV     #15,@MR         ;SET WRAP 3
1249  007444  012777  000061  171036          MOV     #61,@C1         ;LOAD WRITE+GO
1250  007452  012704  040000                  MOV     #40000,R4
1251  007456  005777  171060          LT30B:  TST     @TC             ;SEE IF ALPHA
1252  007462  100015                          BPL     LT30C           ;AWAIT ALPHA
1253  007464  005300                          DEC     R0
1254  007466  001373                          BNE     LT30B
1255  007470  016704  171126                  MOV     EMADDR,R4
1256  007474  004767  014344                  JSR     PC,TTOUT        ;PRINT HEADER
1257  007500  012704  027206                  MOV     #MSG50,R4
1258  007504  004767  014334                  JSR     PC,TTOUT        ;PRINT ALPHA ERROR
1259  007510  004767  013660                  JSR     PC,SCOPE
1260  007514  000435                          BR      LT30X
1261  007516  012777  000015  171010  LT30C:  MOV     #15,@MR         ;CLOCK MR
1262  007524  012777  000015  171002          MOV     #15,@MR         ;CLOCK MR
1263  007532  005000                          CLR     R0
1264  007534  005300                  LT30D:  DEC     R0
1265  007536  001376                          BNE     LT30D           ;DELAY
1266  007540  032777  010000  170756          BIT     #10000,@ER      ;SEE IF DTE SET
1267  007546  001006                          BNE     LT30E           ;IF SO: BR
1268  007550  012767  000001  171140          MOV     #1,EXFL         ;SET EXPT FLG
1269  007556  004767  012224                  JSR     PC,LTGER0       ;GO PRINT ERROR
1270  007562  000412                          BR      LT30X
1271  007564  012701  010000          LT30E:  MOV     #10000,R1       ;SET TEST WORD
1272  007570  017702  170730                  MOV     @ER,R2          ;GET ERROR REGISTER
1273  007574  042702  020000                  BIC     #20000,R2       ;MASK OPI
1274  007600  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
1275  007602  001402                          BEQ     LT30X           ;IF NOT: BR
1276  007604  004767  012164                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
1277  007610  004767  013604          LT30X:  JSR     PC,ITER         ;GO SEE IF ITERATION
1278  007614  004767  011756                  JSR     PC,EORPA        ;GO CLEAR GO BIT
1279  007620  004767  012640                  JSR     PC,DRVCLR
1280  007624  000167  172266                  JMP     TSCD2           ;RETURN TO SCHED
1281
```

```
1282
1283                                        ;LOGIC TEST 31: OPERATION INCOMPLETE(OPI)**********
1284
1285  007630  012767  031202  170764  LT31:   MOV     #MSLT31,EMADDR  ;SET TEST HEADER
1286  007636  012767  007644  171046          MOV     #LT31IT,SCOLP   ;SET SCOPE ADDRESS
1287  007644  012767  000005  170734  LT31IT: MOV     #5,ITAMT        ;SET REDUCED ITER COUNT
1288  007652  004767  013624                  JSR     PC,INIT1        ;INIT, SELECT DRIVE+SLAVE
1289  007656  005000                          CLR     R0
1290  007660  005300                  LT31W:  DEC     R0
1291  007662  001376                          BNE     LT31W           ;AWAIT OPI RESET
1292  007664  052777  000300  170650          BIS     #300,@TC        ;SET FORMAT
1293  007672  012777  000013  170634          MOV     #13,@MR         ;SET WAM 2
1294  007700  005077  170612                  CLR     @FC             ;SET FCS
1295  007704  012777  000061  170576          MOV     #61,@C1         ;LOAD WRITE+GO
1296  007712  012701  020000                  MOV     #20000,R1       ;SET TEST WORD
1297  007716  012703  000004                  MOV     #4,R3
1298  007722  005000                          CLR     R0
1299  007724  032777  020000  170572  LT31A:  BIT     #20000,@ER      ;SEE IF OPI SET
1300  007732  001015                          BNE     LT31B           ;IF SO: BR
1301  007734  005300                          DEC     R0
1302  007736  001372                          BNE     LT31A           ;DELAY
1303  007740  005303                          DEC     R3
1304  007742  001370                          BNE     LT31A
1305  007744  012767  034046  170720          MOV     #TM833A,ERADD   ;SET ERROR CODE
1306  007752  012767  000001  170736          MOV     #1,EXFL         ;SET EXPT FLG
1307  007760  004767  012022                  JSR     PC,LTGER0       ;GO PRINT ERROR
1308  007764  000464                          BR      LT31X
1309  007766  017702  170532          LT31B:  MOV     @ER,R2          ;GET ERROR REGISTER
1310  007772  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
1311  007774  001403                          BEQ     LT31C           ;IF NOT: BR
1312  007776  004767  011772                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
1313  010002  000455                          BR      LT31X
1314  010004  004767  013472          LT31C:  JSR     PC,INIT1        ;GO INIT
1315  010010  005000                          CLR     R0
1316  010012  005300                  LT31W1: DEC     R0
1317  010014  001376                          BNE     LT31W1          ;AWAIT OPI RESET
1318  010016  052777  000300  170516          BIS     #300,@TC        ;SET FORMAT
1319  010024  012777  000015  170502          MOV     #15,@MR         ;SET  RAP 3
1320  010032  012777  000071  170450          MOV     #71,@C1         ;LOAD READ+GO
1321  010040  012701  020000                  MOV     #20000,R1       ;SET TEST WORD
1322  010044  012703  000100                  MOV     #100,R3
1323  010050  005000                          CLR     R0
1324  010052  032777  020000  170444  LT31D:  BIT     #20000,@ER      ;SEE IF OPI SET
1325  010060  001020                          BNE     LT31E           ;IF SO: BR
1326  010062  005300                          DEC     R0
1327  010064  001372                          BNE     LT31D           ;DELAY
1328  010066  005303                          DEC     R3
1329  010070  001370                          BNE     LT31D
1330  010072  012767  010004  170612          MOV     #LT31C,SCOLP    ;SET SCOPE ADDRESS
1331  010100  012767  034062  170564          MOV     #TM833B,ERADD   ;SET ERROR CODE
1332  010106  012767  000001  170602          MOV     #1,EXFL         ;SET EXPT FLG
1333  010114  004767  011666                  JSR     PC,LTGER0       ;GO PRINT ERROR
1334  010120  000406                          BR      LT31X
1335  010122  017702  170376          LT31E:  MOV     @ER,R2          ;GET ERROR REGISTER
1336  010126  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
1337  010130  001402                          BEQ     LT31X           ;IF NOT: BR
```

```
  1338  010132  004767  011636                    JSR     PC,LTGER3       ;ELSE PRINT ERROR
  1339  010136  004767  013256          LT31X:    JSR     PC,ITER         ;GO SEE IF ITERATIONS
  1340  010142  004767  012316                    JSR     PC,DRVCLR
  1341  010146  012767  000020  170432            MOV     #20,ITAMT
  1342  010154  000167  171736                    JMP     TSCD2           ;RETURN TO SCHED
```

```
1343
1344                                        ;LOGIC TEST 32: UNSAFE(UNS)**********
1345
1346  010160  012767  031236  170434  LT32:   MOV     #MSLT32,EMADDR  ;SET TEST HEADER
1347  010166  012767  010174  170516          MOV     #LT32IT,SCOLP   ;SET SCOPE ADDRESS
1348  010174  004767  013302          LT32IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE #SLAVE
1349  010200  016700  170466                  MOV     SLVN,R0         ;GET SLAVE NUMBER
1350  010204  005100                          COM     R0              ;SET NONEXISTANT SLAVE
1351  010206  042700  177770                  BIC     #177770,R0      ;MASK SLAVE NUMBER
1352  010212  052700  000300                  BIS     #300,R0         ;SET FORMAT
1353  010216  010077  170320                  MOV     R0,@TC          ;SELECT ILLEGAL SLAVE
1354  010222  012777  000071  170260          MOV     #71,@C1         ;LOAD READ+GO
1355  010230  012701  040000                  MOV     #40000,R1       ;SET TEST WORD
1356  010234  017702  170264                  MOV     @ER,R2          ;READ ER
1357  010240  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
1358  010242  001011                          BNE     LT32A           ;IF SO: BR
1359  010244  012767  034075  170420          MOV     #TM834,ERADD    ;SET ERROR CODE
1360  010252  012767  000001  170436          MOV     #1,EXFL         ;SET ERROR CODE
1361  010260  004767  011522                  JSR     PC,LTGER0       ;GO PRINT ERROR
1362  010264  000404                          BR      LT32X
1363  010266  020102          LT32A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
1364  010270  001402                          BEQ     LT32X           ;IF NOT: BR
1365  010272  004767  011476                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
1366  010276  004767  013116          LT32X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
1367  010302  004767  012156                  JSR     PC,DRVCLR
1368  010306  000167  171604                  JMP     TSCD2           ;RETURN TO SCHED
```

```
 1369
 1370                                   ;THE FOLLOWING 6 TESTS WILL LOOK AT VARIOUS BITS IN THE
 1371                                   ;DRIVE STATUS(DS) AND TAPE CONTROL(TC)
 1372                                   ;REGISTERS BY FORCING CERTAIN CONDITONS WHICH DO NOT
 1373                                   ;REQUIRE TAPE MOVEMENT.
 1374
 1375                                   ;LOGIC TEST 33: POSITIONING IN PROGRESS(PIP)**********
 1376
 1377  010312  012767  031272  170302  LT33:   MOV     #MSLT33,EMADDR  ;SET TEST HEADER
 1378  010320  012767  010326  170364          MOV     #LT33IT,SCOLP   ;SET SCOPE ADDRESS
 1379  010326  004767  013150          LT33IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE+SLAVE
 1380  010332  012777  000013  170174          MOV     #13,@MR         ;SET WAM 2
 1381  010340  012777  177777  170150          MOV     #-1,@FC         ;SET FCS
 1382  010346  012777  000031  170134          MOV     #31,@C1         ;LOAD SPACE FORWARD+GO
 1383  010354  032777  020000  170140          BIT     #20000,@DS      ;SEE IF PIP=1
 1 84  010362  001010                          BNE     LT33X           ;IF SO: BR
 1385  010364  012767  033655  170300          MOV     #TMS14,ERADD    ;SET ERROR CODE
 1386  010372  012767  000001  170316          MOV     #1,EXFL         ;SET ERROR CODE
 1387  010400  004767  011402                  JSR     PC,LTGER0       ;GO PRINT ERROR
 1388  010404  004767  013010          LT33X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
 1389  010410  000167  171502                  JMP     TSCD2           ;RETURN TO SCHED
```

```
1390
1391                                        ;LOGIC TEST 34: PHASE ENCODED STATUS(PES)********
1392
1393  010414  005767  170242        LT34:   TST     NRZOF           ;SEE IF NRZ ONLY
1394  010420  001054                        BNE     LT34XX          ;IF SO: BR
1395  010422  012767  033575  170242        MOV     #TM86,ERADD     ;SET ERROR CODE
1396  010430  012767  031326  170164        MOV     #MSLT34,EMADDR  ;SET TEST HEADER
1397  010436  012700  000004        LT34IT: MOV     #4,R0
1398  010442  004767  013034        LT34A1: JSR     PC,INIT1        ;GO INIT, SELECT DRIVE+SLAVE
1399  010446  042777  003400  170066        BIC     #3400,@TC       ;SELECT NRZI
1400  010454  032777  000040  170040  LT34A:  BIT     #40,@DS         ;SEE IF PES=0
1401  010462  001410                        BEQ     LT34B           ;IF SO: BR
1402  010464  012767  000002  170224        MOV     #2,EXFL         ;SET RCVD-NOT EXPT
1403  010472  012767  010442  170212        MOV     #LT34A1,SCOLP   ;SET SCOPE ADDRESS
1404  010500  004767  011302                JSR     PC,LTGER0       ;GO PRINT ERROR
1405  010504  062777  000400  170030  LT34B:  ADD     #400,@TC        ;BUMP DENSITY
1406  010512  005300                        DEC     R0              ;SEE IF DONE ALL NRZI
1407  010514  001357                        BNE     LT34A           ;IF NOT: BR
1408  010516  032777  000040  167776  LT34C:  BIT     #40,@DS         ;SEE IF PES=1
1409  010524  001010                        BNE     LT34X           ;IF SO: BR
1410  010526  012767  010516  170156        MOV     #LT34C,SCOLP    ;SET SCOPE ADDRESS
1411  010534  012767  000001  170154        MOV     #1,EXFL         ;SET EXPT-NOT RCVD FLAG
1412  010542  004767  011240                JSR     PC,LTGER0       ;GO PRINT ERROR
1413  010546  004767  012646        LT34X:  JSR     PC,ITER         ;GO SEE IF ITERATION
1414  010552  000167  171340        LT34XX: JMP     T8CD2           ;RETURN TO SCHED
```

```
1415
1416                                                    ;LOGIC TEST 35: TAPE CONTROL WRITE(TCW)**********
1417
1418  010556  012767  034120  170106  LT35:   MOV     #TM837,ERADD
1419  010564  012767  031362  170030          MOV     #M8LT35,EMADDR
1420  010572  004767  012704          LT35IT: JSR     PC,INIT1        ;INIT SELECT DRIVE, SLAVE
1421  010576  032777  000020  167716  18:     BIT     #20,@DS         ;SEE IF SDWN IS RESET
1422  010604  001374                          BNE     18              ;IF NOT: BR
1423  010606  052777  000300  167726          BIS     #300,@TC        ;SET FORMAT
1424  010614  012777  000015  167712          MOV     #15,@WR         ;SET WAM 3
1425  010622  012777  000071  167660          MOV     #71,@C1         ;LOAD READ+GO
1426  010630  032777  020000  167704          BIT     #20000,@TC      ;SEE IF TCW=0
1427  010636  001410                          BEQ     LT35A           ;IF SO: BR
1428  010640  012767  000002  170050          MOV     #2,EXFL         ;SET RCV-NOT EXPT FLAG
1429  010646  012767  010572  170036          MOV     #LT35IT,SCOLP   ;SET SCOPE ADDRESS
1430  010654  004767  011126                  JSR     PC,LTGER0       ;GO PRINT ERROR
1431  010660  004767  012,16          LT35A:  JSR     PC,INIT1        ;INIT
1432  010664  005077  167652                  CLR     @TC             ;WRITE TO TC
1433  010670  032777  020000  167,44          BIT     #20000,@TC      ;SEE IF TCW=1
1434  010676  001010                          BNE     LT35X           ;IF SO: BR
1435  010700  012767  010660  170004          MOV     #LT35A,SCOLP    ;SET SCOPE ADDRESS
1436  010706  012767  000001  170002          MOV     #1,EXFL         ;SE EXPT-NOT RCVD FLAG
1437  010714  004767  011066                  JSR     PC,LTGER0       ;GO PRINT ERROR
1438  010720  004767  012474          LT35X:  JSR     PC,ITER
1439  010724  000167  171166                  JMP     TSCD2           ;RETURN TO SCHED
```

```
1440
1441                                                    ;LOGIC TEST 36: FRAME COUNTER STATUS(FCS)**********
1442
1443  010730  012767  031424  167664  LT36:   MOV     #^RLT36,EMADDR
1444  010736  012767  034126  167726          MOV     #TM838,ERADD    ;SET ERROR CODE
1445  010744  004767  012532          LT36IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE+SLAVE
1446  010750  032777  040000  167564          BIT     #40000,@TC      ;SEE IF FCS=0
1447  010756  001410                          BEQ     LT36A           ;IF SO: BR
1448  010760  012767  010744  167724          MOV     #LT36IT,SCOLP   ;SET SCOPE ADDRESS
1449  010766  012767  000002  167722          MOV     #2,EXFL         ;SET RCVD=NOT EXPT
1450  010774  004767  011006                  JSR     PC,LTGER0       ;GO PRINT ERROR
1451  011000  004767  012476          LT36A:  JSR     PC,INIT1        ;INIT
1452  011004  005077  167506                  CLR     @FC             ;WRITE TO FC
1453  011010  032777  040000  167524          BIT     #40000,@TC      ;SEE IF FCS=1
1454  011016  001010                          BNE     LT36X           ;IF SO: BR
1455  011020  012767  011000  167664          MOV     #LT36A,SCOLP    ;SET SCOPE ADDRESS
1456  011026  012767  000001  167662          MOV     #1,EXFL         ;SET EXPT=NOT RCVD
1457  011034  004767  010746                  JSR     PC,LTGER0       ;GO PRINT ERROR
1458  011040  004767  012354          LT36X:  JSR     PC,ITER
1459  011044  000167  171246                  JMP     TSCD2           ;RETURN TO SCHED
```

```
 1460
 1461                                        ;LOGIC TEST 37: ACCELERATION(ACCL)**********
 1462
 1463  011050  012767  031466  167544  LT37:   MOV     #MSLT37,EMADDR
 1464  011056  012767  034134  167606          MOV     #TM839,ERADD    ;SET ERROR CODE
 1465  011064  004767  012412          LT37IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE+SLAVE
 1466  011070  052777  000300  167444          BIS     #300,@TC        ;SET FORMAT
 1467  011076  005777  167440                  TST     @TC             ;SEE IF ACCL=1
 1468  011102  100410                          BMI     LT37A           ;IF SO: BR
 1469  011104  012767  000001  167604          MOV     #1,EXFL
 1470  011112  012767  011064  167572          MOV     #LT37IT,SCOLP   ;SET SCOPE ADDRESS
 1471  011120  004767  010662                  JSR     PC,LTGER0       ;GO PRINT ERROR
 1472  011124  004767  012352          LT37A:  JSR     PC,INIT1        ;INIT
 1473  011130  052777  000300  167404          BIS     #300,@TC        ;SET FORMAT
 1474  011136  012777  000015  167370          MOV     #15,@MR         ;SET WAM 3
 1475  011144  012777  000071  167336          MOV     #71,@C1         ;LOAD READ+GO
 1476  011152  012700  100000                  MOV     #100000,R0      ;SET ACCL DELAY
 1477  011156  005777  167360          LT37B:  TST     @TC             ;SEE IF ACCL=0
 1478  011162  100012                          BPL     LT37X           ;IF SO: BR
 1479  011164  005300                          DEC     R0
 1480  011166  001373                          BNE     LT37B           ;DELAY
 1481  011170  012767  011124  167514          MOV     #LT37A,SCOLP    ;SET SCOPE ADDRESS
 1482  011176  012767  000002  167512          MOV     #2,EXFL
 1483  011204  004767  010576                  JSR     PC,LTGER0       ;GO PRINT ERROR
 1484  011210  004767  012204          LT37X:  JSR     PC,ITER
 1485  011214  000167  170676                  JMP     TSCD2           ;RETURN TO SCHED
```

```
1486
1487                                        ;LOGIC TEST 40: PE TAPE MARK (TM)**********
1488
1489  011220  005767  167436        LT40:   TST     NRZOF           ;SEE IF NRZ ONLY
1490  011224  001046                        BNE     LT40XX          ;IF SO: BR
1491  011226  012767  011242  167456        MOV     #LT40IT,SCOLP   ;SET SCOPE ADDRESS
1492  011234  012767  031531  167360        MOV     #MSLT40,EMADDR
1493  011242  004767  012234        LT40IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE+SLAVE
1494  011246  005000                        CLR     R0
1495  011250  005300                LT40W:  DEC     R0
1496  011252  001376                        BNE     LT40W           ;DELAY FOR OPI RESET
1497  011254  052777  002300  167260        BIS     #2300,@TC
1498  011262  012777  000007  167244        MOV     #7,@MR          ;SET WAM 0
1499  011270  012777  000027  167212        MOV     #27,@C1         ;LOAD WRITE TAPE MARK+GO
1500  011276  012700  100000                MOV     #100000,R0      ;SET DELAY
1501  011302  032777  000004  167212  LT40A: BIT     #4,@DS          ;SEE IF TM=1
1502  011310  001012                        BNE     LT40X           ;IF SO: BR
1503  011312  005300                        DEC     R0
1504  011314  001372                        BNE     LT40A           ;DELAY
1505  011316  012767  033553  167346        MOV     #TMS3,ERADD
1506  011324  012767  000001  167364        MOV     #1,EXFL
1507  011332  004767  010450                JSR     PC,LTGER0       ;GO PRINT ERROR
1508  011336  004767  012056        LT40X:  JSR     PC,ITER
1509  011342  000167  170550        LT40XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
 1510
 1511                                           ;LOGIC TEST 41: NRZ TAPE MARK (TM,VPE,ITM)**********
 1512
 1513  011346  012767  011362  167336  LT41:   MOV     #LT41IT,SCOLP   ;SET SCOPE ADDRESS
 1514  011354  012767  031576  167240          MOV     #MSLT41,EMADDR
 1515  011362  004767  012114          LT41IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE,SLAVE
 1516  011366  004767  005226                  JSR     PC,BOTT         ;GO ASSURE NOT AT BOT
 1517  011372  052777  001700  167142          BIS     #1700,@TC       ;SET NRZ+NORMAL FORMAT
 1518  011400  012777  177760  167110          MOV     #-20,@FC        ;SET FCS
 1519  011406  012777  000007  167120          MOV     #7,@MR          ;SET WAM 0
 1520  011414  012777  000027  167066          MOV     #27,@C1         ;LOAD WRITE TAPE MARK+GO
 1521  011422  005000                          CLR     R0
 1522  011424  032777  000004  167070  LT41A:  BIT     #4,@DS          ;SEE IF TM=1
 1523  011432  001012                          BNE     LT41B           ;IF SO: BR
 1524  011434  005300                          DEC     R0
 1525  011436  001372                          BNE     LT41A           ;DELAY
 1526  011440  012767  033553  167224          MOV     #TMS3,ERADD     ;SET ERROR CODE
 1527  011446  012767  000001  167242          MOV     #1,EXFL
 1528  011454  004767  010326                  JSR     PC,LTGER0       ;GO PRINT ERROR
 1529  011460  032777  002000  167036  LT41B:  BIT     #2000,@ER       ;SEE IF ITM=1
 1530  011466  001010                          BNE     LT41C           ;IF SO: BR
 1531  011470  012767  034016  167174          MOV     #TMS30,ERADD    ;SET ERROR CODE
 1532  011476  012767  000001  167212          MOV     #1,EXFL
 1533  011504  004767  010276                  JSR     PC,LTGER0       ;GO PRINT ERROR
 1534  011510  032777  000100  167006  LT41C:  BIT     #100,@ER        ;SEE IF VPE=1
 1535  011516  001011                          BNE     LT41D           ;IF SO: BR
 1536  011520  012767  034003  167144          MOV     #TMS28,ERADD    ;SET ERROR CODE
 1537  011526  012767  000001  167162          MOV     #1,EXFL
 1538  011534  004767  010246                  JSR     PC,LTGER0       ;GO PRINT ERROR
 1539  011540  000410                          BR      LT41X
 1540  011542  012701  002100          LT41D:  MOV     #2100,R1        ;SET EXPT ERROR BITS
 1541  011546  017702  166752                  MOV     @ER,R2          ;GET ERROR REGISTER
 1542  011552  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
 1543  011554  001402                          BEQ     LT41X           ;IF NOT: BR
 1544  011556  004767  010212                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
 1545  011562  005002          LT41X:  CLR     R2              ;SET TIMER
 1546  011564  032777  000200  166730  1$:     BIT     #200,@DS        ;SEE IF DRY SET
 1547  011572  001002                          BNE     2$              ;IF SO: BR
 1548  011574  005302                          DEC     R2              ;AWAIT DRY
 1549  011576  001372                          BNE     1$              ;DELAY
 1550  011600  004767  011614          2$:     JSR     PC,ITER         ;GO SEE IF ITERATIONS
 1551  011604  004767  010654                  JSR     PC,DRVCLR       ;GO DO DRIVE CLEAR
 1552  011610  000167  170302                  JMP     TSCD2           ;RETURN TO SCHED
```

```
 1553                                           ;THE FOLLOWING 13 TESTS WILL CHECK DATA FORMATTING
 1554                                           ;AND TRANSFER THROUGH THE TM02 WRAP AROUND MODES
 1555
 1556                                           ;LOGIC TEST 42: WRAP 3, NRZ, NORMAL ODD *****
 1557
 1558  011614  012767  004270  167176  LT42:    MOV     #4270,WCS1      ;SET EXPT CS1
 1559  011622  012767  000100  167172           MOV     #100,WCS2       ;SET EXPT CS2
 1560  011630  012767  010600  167166           MOV     #10600,WDS      ;SET EXPT DS
 1561  011636  012767  000000  167162           MOV     #0,WER          ;SET EXPT ER
 1562  011644  012767  031645  166750           MOV     #MSLT42,EMADDR  ;SET READER
 1563  011652  012767  001700  167116           MOV     #1700,UDES      ;SET NRZ,NORMAL, ODD
 1564  011660  005067  167116          LT42A:   CLR     PATRN           ;POINT TO PATTERN 0
 1565  011664  012767  011672  167020           MOV     #LT42B,SCOLP    ;SET SCOPE ADDRESS
 1566  011672  004767  003706          LT42B:   JSR     PC,WAM3         ;GO DO WRAP 3
 1567  011676  005267  167100           INC     PATRN           ;BUMP PATTERN POINTER
 1568  011702  032767  000004  167072           BIT     #4,PATRN        ;SEE IF DONE
 1569  011710  001770                   BEQ     LT42B           ;IF NOT: BR
 1570  011712  004767  011582           JSR     PC,ITER         ;GO SEE IF ITERATIONS
 1571  011716  005067  167064           CLR     RDRVF           ;CLEAR REVENUE FLAG
 1572  011722  000167  170170           JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
 1573
 1574                                           ;LOGIC TEST 43: WRAP 3, PE, NORMAL, ODD******
 1575
 1576  011726  005767  166738          LT43:   TST     NRZOF           ;SEE IF NRZ ONLY
 1577  011732  001402                          BEQ     LT43A           ;IF NOT: BR
 1578  011734  000167  170156                  JMP     TSCD2           ;RETURN TO SCHED
 1579  011740  012767  004270  167052  LT43A:  MOV     #4270,WC81      ;SET EXPT C81
 1580  011746  012767  000100  167046          MOV     #100,WC82       ;SET EXPT C82
 1581  011754  012767  010640  167042          MOV     #10640,WD8      ;SET EXPT D8
 1582  011762  012767  000000  167036          MOV     #0,WER          ;SET EXPT WER
 1583  011770  012767  031714  166624          MOV     #MSLT43,EMADDR  ;SET HEADER
 1584  011776  012767  002300  166772          MOV     #2300,UDE8      ;SET PE, NORMAL, ODD
 1585  012004  000167  177650                  JMP     LT42A           ;EXECUTE TEST SEQUENCE
```

```
 1586
 1587                                           ;LOGIC TEST 44; WRAP 2, NRZ, NORMAL, ODD*****
 1588
 1589  012010  012767  004260  167002  LT44:    MOV     #4260,WCS1      ;SET EXPT CS1
 1590  012016  012767  000100  166776           MOV     #100,WCS2       ;SET EXPT CS2
 1591  012024  012767  010600  166772           MOV     #10600,WDS      ;SET EXPT DS
 1592  012032  012767  000000  166766           MOV     #0,WER          ;SET EXPT WER
 1593  012040  012767  031762  166554           MOV     #MSLT44,EMADDR  ;SET HEADER
 1594  012046  012767  001700  166722           MOV     #1700,UDES      ;SET TO NRZ,NORMAL, ODD
 1595  012054  005067  166722          LT44A:   CLR     PATRN           ;POINT TO PATTERN 0
 1596  012060  012767  012066  166624           MOV     #LT44B,SCOLP    ;SET SCOPE ADDRESS
 1597  012066  004767  003446          LT44B:   JSR     PC,WAM2         ;GO DO WRAP 2
 1598  012072  005267  166704                   INC     PATRN           ;BUMP POINTER
 1599  012076  032767  000004  166676           BIT     #4,PATRN        ;SEE IF DONE
 1600  012104  001770                           BEQ     LT44B           ;IF NOT; BR
 1601  012106  004767  011306                   JSR     PC,ITER         ;GO SEE IF ITERATIONS
 1602  012112  000167  170000                   JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
 1603
 1604                                                ;LOGIC TEST 45: WRAP 2, PE, NORMAL, ODD*****
 1605
 1606  012116  005767  166540          LT45:  TST     NRZOF           ;SEE IF NRZ ONLY
 1607  012122  001402                        BEQ     LT45A           ;IF NOT: BR
 1608  012124  000167  167766                JMP     TSCD2           ;RETURN TO SCHED
 1609  012130  012767  004260  166662  LT45A: MOV     #4260,WCS1      ;SET EXPT CS1
 1610  012136  012767  000100  166656        MOV     #100,WCS2       ;SET EXPT CS2
 1611  012144  012767  010640  166652        MOV     #10640,WDS      ;SET EXPT DS
 1612  012152  012767  000000  166646        MOV     #0,WER          ;SET EXPT WER
 1613  012160  012767  032031  166434        MOV     #M$LT45,EMADDR  ;SET HEADER
 1614  012166  012767  002300  166602        MOV     #2300,UDES      ;SET PE, NORMAL, ODD
 1615  012174  000167  177654                JMP     LT44A           ;GO EXECUTE TEST SEQUENCES
```

```
1616
1617                                                    ;LOGIC TEST 46: WRAP 1, NRZ, NORMAL, ODD*****
1618
1619  012200  012767  004260  166612  LT46:   MOV     #4260,WC81      ;SET EXPT C81
1620  012206  012767  000100  166606          MOV     #100,WC82       ;SET EXPT C82
1621  012214  012767  010600  166602          MOV     #10600,WDS      ;SET EXPT DS
1622  012222  012767  000000  166576          MOV     #0,WER          ;SET EXPT WER
1623  012230  012767  032077  166364          MOV     #MSLT46,EMADDR  ;SET HEADER
1624  012236  012767  001700  166532          MOV     #1700,UDES      ;SET NRZ, NORMAL, ODD
1625  012244  005067  166532          LT46A:  CLR     PATRN           ;POINT TO PATTERN ZERO
1626  012250  012767  012256  166434          MOV     #LT46B,SCOLP    ;SET SCOPE ADDRESS
1627  012256  004767  003244          LT46B:  JSR     PC,WAM1         ;GO DO WRAP 1
1628  012262  005267  166514                  INC     PATRN           ;BUMP POINTER
1629  012266  032767  000004  166506          BIT     #4,PATRN        ;SEE IF DONE
1630  012274  001770                          BEQ     LT46B           ;IF NOT: BR
1631  012276  004767  011116                  JSR     PC,ITER         ;GO SEE IF ITERATIONS
1632  012302  000167  167610                  JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
 1633
 1634                                           ;LOGIC TEST 47: WRAP 1, PE, NORMAL, ODD*****
 1635
 1636  012306  005767  166350          LT47:   TST     NRZOF           ;SEE IF NRZ ONLY
 1637  012312  001402                          BEQ     LT47A           ;IF NOT: BR
 1638  012314  000167  167576                  JMP     TSCD2           ;RETURN TO SCHED
 1639  012320  004767  007202          LT47A:  JSR     PC,PPGEN        ;GO GENERATE PRE/POSTAMBLE
 1640  012324  012767  004260  166466          MOV     #4260,WCS1      ;SET EXPT CS1
 1641  012332  012767  000100  166462          MOV     #100,WCS2       ;SET EXPT CS2
 1642  012340  012767  010640  166456          MOV     #10640,WDS      ;SET EXPT DS
 1643  012346  012767  000000  166452          MOV     #0,WER          ;SET EXPT WER
 1644  012354  012767  032146  166240          MOV     #MSLT47,EMADDR  ;SET HEADER
 1645  012362  012767  002300  166406          MOV     #2300,UDES      ;SET PE, NORMAL, ODD
 1646  012370  000167  177650                  JMP     LT46A           ;GO EXECUTE TEST SEQUENCE
```

```
1647
1648                                                ;LOGIC TEST 58: WRAP 0, NRZ,NORMAL, ODD*****
1649
1650  012374  012767  144260  166416  LT58:   MOV     #144260,WC81    ;SET EXPT C81
1651  012402  012767  000100  166412          MOV     #100,WC82       ;SET EXPT C82
1652  012410  012767  150600  166406          MOV     #150600,WDS     ;SET EXPT DS
1653  012416  012767  000200  166402          MOV     #200,WER        ;SET EXPT ER
1654  012424  012767  032214  166170          MOV     #MSLT58,EMADDR  ;SET HEADER
1655  012432  012767  001700  166336          MOV     #1700,UDES      ;SET NRZ, NORMAL, ODD
1656  012440  005067  166336          LT58A:  CLR     PATRN           ;POINT TO PATTERN 0
1657  012444  012767  012452  166240          MOV     #LT58B,SCOLP    ;SET SCOPE ADDRESS
1658  012452  004767  003004          LT58B:  JSR     PC,WAM0         ;GO DO WRAP 0
1659  012456  005267  166320                  INC     PATRN           ;BUMP POINTER
1660  012462  032767  000004  166312          BIT     #4,PATRN        ;SEE IF DONE
1661  012470  001770                          BEQ     LT58B           ;IF NOT: BR
1662  012472  004767  010722                  JSR     PC,ITER         ;GO SEE IF ITERATIONS
1663  012476  000167  167414                  JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
 1664
 1665                                           ;LOGIC TEST 51: WRAP 0, PE, NORMAL, ODD*****
 1666
 1667  012502  005767  166154          LT51:   TST     NREOF           ;SEE IF NRE ONLY
 1668  012506  001402                          BEQ     LT51A           ;IF NOT: BR
 1669  012510  000167  167402                  JMP     TSCD2           ;RETURN TI SCHED
 1670  012514  012767  004260  166276  LT51A:  MOV     #4260,WCS1      ;SET EXPT CS1
 1671  012522  012767  000100  166272          MOV     #100,WCS2       ;SET EXPT CS2
 1672  012530  012767  010640  166266          MOV     #10640,WDS      ;SET EXPT DS
 1673  012536  012767  000000  166262          MOV     #0,WER          ;SET EXPT ER
 1674  012544  012767  032263  166050          MOV     #MSLT51,EMADDR  ;SET HEADER
 1675  012552  012767  002300  166216          MOV     #2300,UDES      ;SET PE, NORMAL, ODD
 1676  012560  000167  177654                  JMP     LTS0A           ;GO EXECUTE TEST SEQUENCE
```

```
1677
1678                                                ;LOGIC TEST 52: CORE DUMP WRITE, WAM2********
1679
1680  012564  012767  004260  166226  LT52:   MOV     #4260,WC81      ;SET EXPT C81
1681  012572  012767  000100  166222          MOV     #100,WC82       ;SET EXPT C82
1682  012600  012767  010600  166216          MOV     #10600,WD8      ;SET EXPT D8
1683  012606  012767  000000  166212          MOV     #0,WER          ;SET EXPT ER
1684  012614  012767  032331  166000          MOV     #M8LT52,EMADDR  ;SET HEADER
1685  012622  012767  001720  166146          MOV     #1720,UDE8      ;SET NRZ, CORE DUMP, ODD
1686  012630  005067  166146                  CLR     PATRN           ;POINT TO PATTERN 0
1687  012634  012767  012642  166050          MOV     #LT52A,SCOLP    ;SET SCOPE ADDRESS
1688  012642  004767  002672          LT52A:  JSR     PC,WAM2         ;GO DO WAM 2
1689  012646  022767  000002  166126          CMP     #2,PATRN        ;SEE IF DONE
1690  012654  001404                          BEQ     LT52X           ;IF SO: BR
1691  012656  012767  000002  166116          MOV     #2,PATRN        ;SELECT PATTERN 2
1692  012664  000766                          BR      LT52A           ;CONTINUE
1693  012666  004767  010526          LT52X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
1694  012672  000167  167220                  JMP     TSCD2           ;RETURN TO SCHEDULES
1695
```

```
 1696
 1697                                                ;LOGIC TEST 53: CORE DUMP READ, WAM 3*****
 1698
 1699  012676  012767  004270  166114  LT53:   MOV     #4270,WCS1      ;SET EXPT CS1
 1700  012704  012767  000100  166110          MOV     #100,WCS2       ;SET EXPT CS2
 1701  012712  012767  010600  166104          MOV     #10600,WDS      ;SET EXPT DS
 1702  012720  012767  000000  166100          MOV     #0,WER          ;SET EXPT ER
 1703  012726  012767  032402  165666          MOV     #M&LT53,EMADDR  ;SET HEADER
 1704  012734  012767  001720  166034          MOV     #1720,UDES      ;SELECT NRE, CORE DUMP, ODD
 1705  012742  005067  166034                  CLR     PATRN           ;SELECT PATTERN 0
 1706  012746  012767  012762  165736          MOV     #LT53A,SCOLP    ;SET SCOPE ADDRESS
 1707  012754  012767  001054  166026          MOV     #WCDP0,RCDP     ;POINT TO PATTERN 0
 1708  012762  004767  002616          LT53A:  JSR     PC,WAM3         ;GO DO WAM3
 1709  012766  022767  000002  166006          CMP     #2,PATRN        ;SEE IF DONE
 1710  012774  001407                          BEQ     LT53X           ;IF SO: BR
 1711  012776  012767  000002  165776          MOV     #2,PATRN        ;SELECT PATTERN 2
 1712  013004  012767  001042  165776          MOV     #WCDP2,RCDP     ;POINT TO PATTERN 2
 1713  013012  000763                          BR      LT53A           ;CONTINUE
 1714  013014  004767  010400          LT53X:  JSR     PC,ITER         ;GO SEE IF ITERATION
 1715  013020  000167  167072                  JMP     TSCD2           ;RETURN TO SCHEDULE
```

```
 1716
 1717                                           ;LOGIC TEST 54; EVEN PARITY WRITE; WAM 1(M8903)*****
 1718
 1719  013024  012767  004260  165766  LT54:   MOV     #4260,WC81      ;SET EXPT C81
 1720  013032  012767  000100  165762          MOV     #100,WC82       ;SET EXPT C82
 1721  013040  012767  010600  165756          MOV     #10600,WDS      ;SET EXPT DS
 1722  013046  012767  000000  165752          MOV     #0,WER          ;SET EXPT ER
 1723  013054  012767  032452  165540          MOV     #MSLT54,EMADDR  ;SET HEADER
 1724  013062  012767  001710  165706          MOV     #1710,UDES      ;SET NRZ, NORMAL, EVEN
 1725  013070  000167  177150                  JMP     LT46A           ;GO EXECUTE WAM 1
```

```
 1726
 1727                                          ;LOGIC TEST 55: EVEN PARITY READ: WAM 0(M8983 M8984)*****
 1728
 1729  013074  012767  144260  165716  LT55:  MOV     #144260,WC81     ;SET EXPT C81
 1730  013102  012767  000100  165712         MOV     #100,WC82        ;SET EXPT C82
 1731  013110  012767  150600  165706         MOV     #150600,WD8      ;SET EXPT D8
 1732  013116  012767  000200  165702         MOV     #200,WER         ;SET EXPT ER
 1733  013124  012767  032533  165670         MOV     #M8LT55,EMADDR   ;SET HEADER
 1734  013132  012767  001710  165636         MOV     #1710,UDE8       ;SET NRZ, NORMAL, EVEN
 1735  013140  000167  177274                 JMP     LT50A            ;GO DO WAM 0
```

```
 1736
 1737                                          ;LOGIC TEST 56: READ REVERSE: WAM 3(M8906)*****
 1738
 1739  013144  012767  004276  165646  LT56:   MOV     #4276,WC81      ;SET EXPT C81
 1740  013152  012767  000100  165642          MOV     #100,WC82       ;SET EXPT C82
 1741  013160  012767  010640  165636          MOV     #10640,WD8      ;SET EXPT D8
 1742  013166  012767  000000  165632          MOV     #0,WER          ;SET EXPT ER
 1743  013174  012767  032612  165420          MOV     #M8LT56,EMADDR  ;SET HEADER
 1744  013202  012767  002300  165566          MOV     #2300,UDE8      ;SELECT PE,NORMAL,ODD
 1745  013210  012767  000001  165570          MOV     #1,RDRVF        ;SET READ REVERSE FLAG
 1746  013216  000167  176436                  JMP     LT42A           ;GO DO WAM 3, REVERSE
 1747
```

```
 1748
 1749                                   ;THE FOLLOWING SIX(6) TEST WILL REQUIRE TAPE MOVEMENT.  EACH
 1750                                   ;TEST WILL PERFORM A TAPE WRITE WHILE IN A PARITICULAR MAINTENANCE
 1751                                   ;MODE IN ORDER TO FORCE THE REMAINING ERROR CONDITIONS.
 1752
 1753                                   ;LOGIC TEST 57: CYCLIC REDUNDANCY ERROR(CRC)**********
 1754
 1755  013222  004767  001742          LT57:   JSR     PC,$TATIC       ;GO SEE IF STATIC ONLY
 1756  013226  012700  001000                  MOV     #1000,R0
 1757  013232  005300                  LT57PS: DEC     R0
 1758  013234  001376                          BNE     LT57PS          ;PAUSE
 1759  013236  012767  032657  165356          MOV     #M$LT57,EMADDR
 1760  013244  012767  013252  165440          MOV     #LT57IT,$COLP   ;SET SCOPE ADDRESS
 1761  013252  004767  010224          LT57IT: JSR     PC,INIT1        ;INIT SELECT DRIVE+SLAVE
 1762  013256  052777  001700  165256          BIS     #1700,@TC       ;SET NRZ + NORMAL FORMAT
 1763  013264  012777  177770  165220          MOV     #-10,@WC
 1764  013272  012777  177760  165216          MOV     #-20,@FC        ;SET FC=20
 1765  013300  012777  034144  165206          MOV     #WDATA,@BA      ;SET BUS ADDRESS
 1766  013306  012777  000007  165220          MOV     #7,@MR          ;SET MM CODE
 1767  013314  012777  000061  165166          MOV     #61,@C1         ;LOAD WRITE+GO
 1768  013322  005000                          CLR     R0
 1769  013324  032777  000200  165170  LT57A:  BIT     #200,@DS        ;SEE IF DRY=1
 1770  013332  001002                          BNE     LT57B           ;IF SO: BR
 1771  013334  005300                          DEC     R0
 1772  013336  001372                          BNE     LT57A           ;DELAY
 1773  013340  022777  000200  165156  LT57B:  CMP     #200,@ER        ;SEE IF LRC ERROR ONLY
 1774  013346  001007                          BNE     LT57B1          ;IF NOT: BR
 1775  013350  017702  165154                  MOV     @CC,R2          ;GET CHECK CHAR
 1776  013354  042702  177000                  BIC     #177000,R2      ;MASK CRC
 1777  013360  022702  000777                  CMP     #777,R2         ;SEE IF SETUP CRC IS CORRECT
 1778  013364  001410                          BEQ     LT57B2          ;IF SO: BR
 1779  013366  004767  006402          LT57B1: JSR     PC,LTGER3       ;ELSE PRINT ERROR SETUP
 1780  013372  012704  027313                  MOV     #M$G55,R4
 1781  013376  004767  010442                  JSR     PC,TTOUT        ;PRINT SITUP ERROR MSG
 1782  013402  000167  166510                  JMP     TSCD2           ;RETURN TO SCHED
 1783  013406  004767  010070          LT57B2: JSR     PC,INIT1        ;GO INIT
 1784  013412  052777  000300  165122          BIS     #300,@TC        ;SET FORMAT+NRZ
 1785  013420  012777  177770  165064          MOV     #-10,@WC        ;SET WC
 1786  013426  012777  177760  165062          MOV     #-20,@FC        ;SET FC
 1787  013434  012777  034144  165052          MOV     #WDATA,@BA      ;SET BA
 1788  013442  012777  000021  165064          MOV     #21,@MR         ;SET MM
 1789  013450  012777  000061  165032          MOV     #61,@C1         ;LOAD WRITE+GO
 1790  013456  005000                          CLR     R0
 1791  013460  032777  000200  165034  LT57C:  BIT     #200,@DS        ;SEE IF DRY
 1792  013466  001002                          BNE     LT57D           ;IF SO: BR
 1793  013470  005300                          DEC     R0
 1794  013472  001372                          BNE     LT57C           ;AWAIT DRY
 1795  013474  005777  165024          LT57D:  TST     @ER             ;SEE IF CRC=1
 1796  013500  100411                          BMI     LT57E           ;IF SO: BR
 1797  013502  012767  034112  165162          MOV     #TM836,ERADD    ;SET ERROR CODE
 1798  013510  012767  000001  165200          MOV     #1,EXFL
 1799  013516  004767  006264                  JSR     PC,LTGER0       ;GO PRINT ERROR
 1800  013522  000410                          BR      LT57X
 1801  013524  012701  100200          LT57E:  MOV     #100200,R1      ;SET EXPT ERROR BITS
 1802  013530  017702  164770                  MOV     @ER,R2          ;GET ERROR REGISTER
 1803  013534  020102                          CMP     R1,R2           ;SEE IF EXPECTED
```

```
 1004  013536  001402                    BEQ     LTS7X           ;IF NOT; BR
 1005  013540  004767  006230            JSR     PC,LTGER3       ;ELSE PRINT ERROR
 1006  013544  004767  007650    LTS7X:  JSR     PC,ITER         ;DO ITERATIONS
 1007  013550  004767  006710            JSR     PC,DRVCLR
 1008  013554  000167  166336            JMP     TSCD2           ;RETURN TO SCHED
```

```
 1809
 1810                                          ;LOGIC TEST 60: LONGITUALINAL REDUNDANCY(LRC)**********
 1811
 1812  013560  004767  001404          LT60:   JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
 1813  013564  012767  013600  165120          MOV     #LT60IT,SCOLP   ;SET SCOPE ADDRESS
 1814  013572  012767  032713  165022          MOV     #MSLT60,EMADDR
 1815  013600  004767  007676          LT60IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE+SLAVE
 1816  013604  052777  000300  164730          BIS     #300,@TC        ;SET FORMAT+NRZ
 1817  013612  012777  000023  164714          MOV     #23,@MR         ;SET MM
 1818  013620  012777  177770  164664          MOV     #-10,@WC        ;SET WC
 1819  013626  012777  177760  164662          MOV     #-20,@FC        ;SET FC
 1820  013634  012777  034144  164652          MOV     #WDATA,@BA      ;SET BA
 1821  013642  012777  000061  164640          MOV     #61,@C1         ;LOAD WRITE+GO
 1822  013650  005000                          CLR     R0
 1823  013652  032777  000200  164642  LT60C:  BIT     #200,@DS        ;SEE IF DRY
 1824  013660  001002                          BNE     LT60D           ;IF SO: BR
 1825  013662  005300                          DEC     R0
 1826  013664  001372                          BNE     LT60C           ;AWAIT DRY
 1827  013666  032777  000200  164630  LT60D:  BIT     #200,@ER        ;SEE IF LRC=1
 1828  013674  001011                          BNE     LT60E           ;IF SO: BR
 1829  013676  012767  033767  164766          MOV     #TM826,ERADD    ;SET ERROR CODE
 1830  013704  012767  000001  165004          MOV     #1,EXFL
 1831  013712  004767  006070                  JSR     PC,LTGER0       ;GO PRINT
 1832  013716  000425                          BR      LT60X
 1833  013720  017702  164610          LT60E:  MOV     @MR,R2
 1834  013724  042702  000177                  BIC     #177,R2         ;MASK LRC
 1835  013730  012701  157600                  MOV     #157600,R1      ;SET EXPT LRC
 1836  013734  020102                          CMP     R1,R2           ;SEE IF EXPT = RCVD
 1837  013736  001405                          BEQ     LT60F           ;IF SO: BR
 1838  013740  012767  027270  164724          MOV     #MSG53,ERADD    ;SET ERROR CODE
 1839  013746  004767  007202                  JSR     PC,LTGER1       ;PRINT ERROR
 1840  013752  017702  164546          LT60F:  MOV     @ER,R2          ;GET ERROR REGISTER
 1841  013756  012701  000200                  MOV     #200,R1         ;SET EXPT ERROR BITS
 1842  013762  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
 1843  013764  001402                          BEQ     LT60X           ;IF NOT: BR
 1844  013766  004767  006002                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
 1845  013772  004767  007422          LT60X:  JSR     PC,ITER
 1846  013776  004767  006462                  JSR     PC,DRVCLR
 1847  014002  000167  166110                  JMP     TSCD2           ;RETURN TO SCHED
```

```
1848
1849                                                ;LOGIC TEST 61: PE CORRECTABLE DATA (CORR)**********
1850
1851  014006  005767  164650          LT61:   TST     MRIOF           ;SEE IF MRI ONLY
1852  014012  001122                          BNE     LT61XX          ;IF SO: BR
1853  014014  004767  001150                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
1854  014020  012767  032747  164574          MOV     #MSLT61,EMADDR
1855  014026  012767  014034  164556          MOV     #LT61IT,SCOLP
1856  014034  004767  007442          LT61IT: JSR     PC,INIT1        ;INIT, SELCT DRIVE+SLAVE
1857  014040  052777  002300  164474          BIS     #2300,@TC       ;SET PE,NORMAL
1858  014046  012777  177600  164436          MOV     #-200,@WC       ;SET WC=200
1859  014054  012777  177400  164434          MOV     #-400,@FC       ;SET FC=400
1860  014062  012777  034144  164424          MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
1861  014070  012777  000061  164412          MOV     #61,@C1         ;LOAD WRITE+GO
1862  014076  005000                          CLR     R0
1863  014100  005777  164412          LT61A:  TST     @FC             ;SEE IF FC=0
1864  014104  001402                          BEQ     LT61A1          ;IF SO: BR
1865  014106  005300                          DEC     R0
1866  014110  001373                          BNE     LT61A           ;DELAY FOR FC=0
1867  014112  012777  000021  164414  LT61A1: MOV     #21,@MR         ;SET MAINT CODE
1868  014120  005000                          CLR     R0
1869  014122  032777  000200  164372  LT61B:  BIT     #200,@DS        ;SEE IF DRY IS SET
1870  014130  001002                          BNE     LT61C           ;IF SO: BR
1871  014132  005300                          DEC     R0
1872  014134  001372                          BNE     LT61B           ;AWAIT DRY
1873  014136  005777  164362          LT61C:  TST     @ER             ;SEE IF CORR=1
1874  014142  100410                          BMI     LT61D           ;IF SO: BR
1875  014144  012767  034103  164520          MOV     #TM835,ERADD    ;SET ERROR CODE
1876  014152  012767  000001  164536          MOV     #1,EXFL
1877  014160  004767  005622                  JSR     PC,LTGER0       ;GO PRINT ERROR
1878  014164  000240                  LT61D:  NOP
1879  014166  000240                  LT61E:  NOP
1880  014170  122777  000002  164332          CMPB    #2,@CC          ;SEE IF DEAD TRACK BIT 1
1881  014176  001414                          BEQ     LT61F           ;IF SO: BR
1882  014200  117702  164324                  MOVB    @CC,R2          ;SAVE RCVD
1883  014204  042702  177000                  BIC     #177000,R2      ;MASK OUT CRC
1884  014210  112701  000002                  MOVB    #2,R1           ;SAVE EXPT
1885  014214  012767  026663  164450          MOV     #MSG42,ERADD    ;SET ERROR CODE
1886  014222  004767  006726                  JSR     PC,LTGER1       ;GO PRINT ERROR
1887  014226  000410                          BR      LT61X
1888  014230  017702  164270          LT61F:  MOV     @ER,R2          ;GET ERROR REGISTER
1889  014234  012701  100000                  MOV     #100000,R1      ;SET EXPT ERROR BITS
1890  014240  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
1891  014242  001402                          BEQ     LT61X           ;IF NOT: BR
1892  014244  004767  005524                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
1893  014250  004767  007144          LT61X:  JSR     PC,ITER
1894  014254  004767  006204                  JSR     PC,DRVCLR
1895  014260  000167  165632          LT61XX: JMP     TSCD2           ;RETURN TO SCHED
1896
```

```
1897
1898                                        ;LOGIC TEST 62: PE INCORRECTABLE DATA(INC,)**********
1899
1900  014264  005767  164372        LT62:   TST     NRIOF           ;SEE IF NRI ONLY
1901  014270  001120                        BNE     LT62XX          ;IF SO: BR
1902  014272  004767  000672                JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
1903  014276  012767  033027  164316        MOV     #MSLT62,EMADDR
1904  014304  012767  014312  164400        MOV     #LT62IT,SCOLP
1905  014312  004767  007164        LT62IT: JSR     PC,INIT1        ;INIT SELECT DRIVE SLAVE
1906  014316  012777  177600  164166        MOV     #-200,@WC       ;SET WC=200
1907  014324  012777  177400  164164        MOV     #-400,@FC       ;SET FC=400
1908  014332  012777  034144  164154        MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
1909  014340  052777  002300  164174        BIS     #2300,@TC       ;SET TO PE,NORMAL
1910  014346  012777  000061  164134        MOV     #61,@C1         ;LOAD WRITE+GO
1911  014354  005000                        CLR     R0
1912  014356  005777  164134        LT62E:  TST     @FC             ;AWAIT FC=0
1913  014362  001402                        BEQ     LT62E1
1914  014364  005300                        DEC     R0
1915  014366  001373                        BNE     LT62E           ;AWAIT FC=0
1916  014370  012777  000023  164136  LT62E1: MOV     #23,@MR         ;SET MAINT CODE
1917  014376  005000                        CLR     R0
1918  014400  032777  000200  164114  LT62A:  BIT     #200,@DS        ;SEE IF DRY IS SET
1919  014406  001002                        BNE     LT62B           ;IF SO: BR
1920  014410  005300                        DEC     R0
1921  014412  001372                        BNE     LT62A           ;AWAIT DRY
1922  014414  032777  000100  164102  LT62B:  BIT     #100,@ER        ;SEE IF INC=1
1923  014422  001010                        BNE     LT62D           ;IF SO:BR
1924  014424  012767  033745  164240        MOV     #TM823,ERADD    ;SET ERROR CODE
1925  014432  012767  000001  164256        MOV     #1,EXPL
1926  014440  004767  005342                JSR     PC,LTGER0       ;GO PRINT ERROR
1927  014444  017702  164060        LT62D:  MOV     @CC,R2          ;GET CHECK CHAR
1928  014450  042702  177000                BIC     #177000,R2      ;MASK CHECK CHAR
1929  014454  012701  000046                MOV     #46,R1          ;SET EXPT CK
1930  014460  020102                        CMP     R1,R2           ;SEE IF EXPT = RCVD
1931  014462  001405                        BEQ     LT62F           ;IF SO: BR
1932  014464  012767  027302  164200        MOV     #MSG54,ERADD
1933  014472  004767  006456                JSR     PC,LTGER1       ;ELSE GO PRINT ERROR
1934  014476  017702  164022        LT62F:  MOV     @ER,R2
1935  014502  042702  100600                BIC     #100600,R2      ;MASK NSG AND CORR
1936  014506  012701  000100                MOV     #100,R1         ;SET EXPT ERROR BITS
1937  014512  020102                        CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
1938  014514  001402                        BEQ     LT62X           ;IF NOT: BR
1939  014516  004767  005252                JSR     PC,LTGER3       ;ELSE PRINT ERROR
1940  014522  004767  006672        LT62X:  JSR     PC,ITER
1941  014526  004767  005732                JSR     PC,DRVCLR
1942  014532  000167  165360        LT62XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
                                        ;LOGIC TEST 63: PE FORMAT ERROR(PEF,NSG)**********
014536  005767  164120          LT63:   TST     NRIOF           ;SEE IF NRI ONLY
014542  001114                          BNE     LT63XX          ;IF SO: BR
014544  004767  000420                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
014550  012767  033111  164044          MOV     #MSLT63,EMADDR  ;SET HEADER
014556  012767  014564  164126          MOV     #LT63IT,SCOLP   ;SET SCOPE ADDRESS
014564  004767  006712          LT63IT: JSR     PC,INIT1        ;INITIALIZE
014570  012777  177770  163714          MOV     #-10,@WC        ;SET WC=10
014576  012777  177760  163712          MOV     #-20,@FC        ;SET FC=20
014604  052777  002300  163730          BIS     #2300,@TC       ;SET TO PE,NORMAL
014612  012777  034144  163674          MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
014620  012777  000061  163662          MOV     #61,@C1         ;LOAD WRITE+GO
014626  005777  163664          LT63A:  TST     @FC
014632  001375                          BNE     LT63A           ;AWAIT FC 0
014634  032777  000100  163672  10:     BIT     #100,@MR
014642  001774                          BEQ     10              ;TIME DELAY
014644  032777  000100  163662  20:     BIT     #100,@MR
014652  001374                          BNE     20
014654  032777  000100  163652  30:     BIT     #100,@MR
014662  001774                          BEQ     30
014664  012777  000027  163642          MOV     #27,@MR         ;SET MM CODE TO KILL PEF
014672  012700  004000                  MOV     #4000,R0
014676  032777  000200  163616  LT63B:  BIT     #200,@DS        ;SEE IF DRY SET
014704  001002                          BNE     LT63C           ;IF SO: BR
014706  005300                          DEC     R0
014710  001372                          BNE     LT63B           ;AWAIT DRY
014712  032777  000200  163604  LT63C:  BIT     #200,@ER        ;SEE IF PEF SET
014720  001011                          BNE     LT63D           ;IF SO: BR
014722  012767  033761  163742          MOV     #TM525,ERADD    ;SET ERROR TAG
014730  012767  000001  163760          MOV     #1,EXFL         ;SET EXPT FLAG
014736  004767  005044                  JSR     PC,LTGER0       ;GO PRINT ERROR
014742  000410                          BR      LT63X
014744  017702  163554          LT63D:  MOV     @ER,R2          ;GET ERROR REGISTER
014750  012701  000600                  MOV     #600,R1         ;SET EXPT ERROR BITS
014754  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
014756  001402                          BEQ     LT63X           ;IF NOT: BR
014760  004767  005010                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
014764  004767  006430          LT63X:  JSR     PC,ITER
014770  004767  005470                  JSR     PC,DRVCLR
014774  000167  165116          LT63XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
 1985                                                   ;LOGIC TEST 64: FRAME COUNT OVERFLOW(M8985)*******
 1986
 1987  015000  012767  033145  163614  LT64:   MOV     #MSLT64,EMADDR  ;SET TEST HEADER
 1988  015006  012767  015014  163676          MOV     #LT64IT,SCOLP   ;SET SCOPE ADDRESS
 1989  015014  004767  006462          LT64IT: JSR     PC,INIT1        ;GO INIT
 1990  015020  012777  177770  163464          MOV     #-10,@WC        ;SET WC = 10
 1991  015026  012777  177760  163462          MOV     #-20,@FC        ;SET FC = 20
 1992  015034  052777  001700  163500          BIS     #1700,@TC       ;SET TO NRZ, NORMAL, ODD
 1993  015042  012777  034144  163444          MOV     #WDATA,@BA      ;SET BUS ADDRESS
 1994  015050  012777  000013  163456          MOV     #13,@MR         ;SET WRAP 2
 1995  015056  012777  000061  163424          MOV     #61,@C1         ;LOAD WRITE+GO
 1996  015064  012700  040000                  MOV     #40000,R0
 1997  015070  005777  163446          LT64A:  TST     @TC             ;SEE IF ALPHA
 1998  015074  100002                          BPL     LT64B           ;IF SO: BR
 1999  015076  005300                          DEC     R0
 2000  015100  001373                          BNE     LT64A           ;AWAIT ALPHA
 2001  015102  012700  000020          LT64B:  MOV     #20,R0          ;SET CLK CNT
 2002  015106  052777  000040  163420  LT64C:  BIS     #40,@MR
 2003  015114  042777  000040  163412          BIC     #40,@MR         ;CLOCK MR
 2004  015122  005300                          DEC     R0
 2005  015124  001370                          BNE     LT64C           ;IF NOT DONE ALL: BR
 2006  015126  017702  163364                  MOV     @FC,R2
 2007  015132  005001                          CLR     R1              ;SET TEST WORD
 2008  015134  020102                          CMP     R1,R2           ;SEE IF EXPT = RCVD
 2009  015136  001410                          BEQ     LT64X           ;IF SO: BR
 2010  015140  012767  026136  163524          MOV     #MSG19,ERADD    ;SET ERROR CODE
 2011  015146  012767  000001  163542          MOV     #1,EXFL         ;SET EXPT FLAG
 2012  015154  004767  005774                  JSR     PC,LTGER1       ;GO PRINT ERROR
 2013  015160  004767  006234          LT64X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
 2014  015164  000167  164726                  JMP     TSCD2           ;RETURN TO SCHEDULAR
 2015
 2016                                                   ;STATIC TESTS ONLY SUBROUTINE*********
 2017
 2018  015170  005767  163544          STATIC: TST     STFLG           ;SEE IF SINGLE TEST ONLY
 2019  015174  001006                          BNE     STATX           ;IF SO: BR
 2020  015176  005767  163610                  TST     STATC           ;SEE IF STATIC ONLY
 2021  015202  001403                          BEQ     STATX           ;IF NOT: BR
 2022  015204  005726                          TST     (SP)+           ;RESET STACK
 2023  015206  000167  164704                  JMP     TSCD2           ;RETURN TO SCHEDULAR
 2024  015212  000207                  STATX:  RTS     PC              ;RETURN TO TEST
 2025
```

```
2026
2027                                    ;*****************************************************
2028                                    ;DATA SETUP ROUTINE:
2029                                    ;
2030                                    ;THIS ROUTINE IS USED TO GENERATE THE DATA PATTERNS
2031                                    ;USED BY THE WRAP AROUND TESTS (LT50-LT64).
2032                                    ;*****************************************************
2033
2034  015214  005767  163520      DSUP:  TST     STFLG           ;SEE IF SINGLE TEST
2035  015220  001431                     BEQ     D80             ;IF NOT: BR
2036  015222  032777  000100  163340     BIT     #100,@SWR       ;SEE IF SELECT PATTERN
2037  015230  001425                     BEQ     D80             ;IF NOT: BR
2038  015232  012704  034562             MOV     #WMSG3,R4
2039  015236  004767  006602             JSR     PC,TTOUT        ;REQUEST PATTERN NUMBER
2040  015242  016703  163534             MOV     PATRN,R3
2041  015246  004767  006734             JSR     PC,OCTP         ;PRINT PATTERN NUMBER
2042  015252  012705  001002             MOV     #PATRN,R5       ;GET ADDRESS OF PATRN ENTRY
2043  015256  012701  000001             MOV     #1,R1           ;SET SIZE OF ENTRY
2044  015262  012702  000003             MOV     #3,R2           ;SET UPPER LIMIT
2045  015266  012703  000000             MOV     #0,R3           ;SET LOWER LIMIT
2046  015272  004767  006310             JSR     PC,TTR          ;GO GET PATTERN NUMBER
2047  015276  012767  000001  163474     MOV     #1,WPGFL        ;SET FLAG
2048  015304  012703  035472      D80:   MOV     #WBUFF,R3       ;R3 = ADDRS OF WRITE BUFFER
2049  015310  016701  163466             MOV     PATRN,R1        ;R1 = PATTERN SELECTOR
2050  015314  062701  000001             ADD     #1,R1           ;BUMP POINTER
2051  015320  000241                     CLC
2052  015322  006101                     ROL     R1              ;MAKE PATTERN SELECTOR EVEN
2053  015324  000171  001030             JMP     @DATBL(R1)      ;GO GENERATE PATTERN
2054  015330  032777  010000  163200     BIT     #10000,@DT      ;SEE IF SEVEN TRACK
2055  015336  001410                     BEQ     D83             ;IF NOT: BR
2056  015340  012702  000202             MOV     #202,R2         ;SET BUFFER SIZE
2057  015344  012701  035472             MOV     #WBUFF,R1       ;SET START OF BUFFER
2058  015350  042721  140300      D82:   BIC     #140300,(R1)+   ;MASK FOR 7 CH
2059  015354  005302                     DEC     R2              ;SEE IF DONE
2060  015356  001374                     BNE     D82             ;IF NOT: BR
2061  015360  012702  000202      D83:   MOV     #202,R2         ;R2=BUFFER SIZE +2
2062  015364  012701  036104             MOV     #RBUFF,R1       ;R1=READ DATA START
2063  015370  005021              D84:   CLR     (R1)+           ;CLEAR BUFFER
2064  015372  005302                     DEC     R2              ;SEE IF DONE ALL
2065  015374  001375                     BNE     D84             ;IF NOT: BR
2066  015376  000207                     RTS     PC              ;EXIT
2067
2068                                    ;ALL ONES**********
2069
2070  015400  012701  177777      DAT1:  MOV     #-1,R1          ;R1=DATA
2071  015404  012702  000202      DAT1A: MOV     #202,R2         ;R2=WORD COUNT +2
2072  015410  010123              DAT1B: MOV     R1,(R3)+        ;LOAD BUFFER
2073  015412  005302                     DEC     R2              ;SEE IF DONE
2074  015414  001375                     BNE     DAT1B           ;IF NOT: BR
2075  015416  000167  177706             JMP     D81             ;RETURN
2076
```

```
2077
2078                                    ;ALL ZEROS**********
2079
2080  015422  005001            DAT2:   CLR     R1              ;R1=DATA
2081  015424  000167  177754            JMP     DAT1A           ;LOAD BUFFER
2082
2083                                    ;ONE/ZERO IN ALTERNATING CHARACTERS**********
2084
2085  015430  012701  125125    DAT3:   MOV     #125125,R1      ;R1=DATA
2086  015434  000167  177744            JMP     DAT1A           ;LOAD BUFFER
2087
2088                                    ;ALL BITS 0-377**********
2089
2090  015440  005001            DAT4:   CLR     R1              ;R1=STARTING DATA
2091  015442  012702  000404            MOV     #404,R2         ;R2=CHARACTER COUNT
2092  015446  110123            DAT4A:  MOVB    R1,(R3)+        ;LOAD BUFFER
2093  015450  105201                    INCB    R1              ;BUMP DATA
2094  015452  005302                    DEC     R2              ;SEE IF DONE
2095  015454  001374                    BNE     DAT4A           ;IF NOT: BR
2096  015456  000167  177646            JMP     D81             ;RETURN
2097
```

```
2098
2099                                            ;WRAP AROUND MODE 0 GLOBAL*****
2100
2101  015462  012767  000006  163260  WAM0:   MOV     #6,WAM          ;SET WAM NUMBER
2102  015470  012767  000060  163254  WAM01:  MOV     #60,FUN
2103  015476  005067  163252                  CLR     DATC
2104  015502  012767  035472  163250          MOV     #WBUFF,DATAD    ;SET BUFFER ADDRESS
2105  015510  012767  036104  163244          MOV     #RBUFF,RDAD     ;SET POINTER TO READ BUFFER
2106  015516  004767  000170                  JSR     PC,SETUP        ;GO SET UP
2107  015522  000167  000516                  JMP     EXEC
2108
2109                                            ;WRAP AROUND MODE 1 WRITE BUFFER**********
2110
2111  015526  012767  000010  163214  WAM1:   MOV     #10,WAM
2112  015534  000167  177730                  JMP     WAM01
2113
2114                                            ;WRAP AROUND MODE 2 BIT FIDDLER WRITE**********
2115
2116  015540  012767  000012  163202  WAM2:   MOV     #12,WAM
2117  015546  012767  000060  163176          MOV     #60,FUN
2118  015554  005067  163174                  CLR     DATC
2119  015560  012767  035472  163172          MOV     #WBUFF,DATAD
2120  015566  012767  036104  163166          MOV     #RBUFF,RDAD
2121  015574  004767  000112          WAM2A:  JSR     PC,SETUP
2122  015600  000167  000440                  JMP     EXEC
2123
2124                                            ;WRAP AROUND MODE 3 BIT FIDDLER READ**********
2125
2126  015604  012767  000014  163136  WAM3:   MOV     #14,WAM         ;SET WAM NUMBER
2127  015612  012767  000070  163132          MOV     #70,FUN         ;SET FUNCTION
2128  015620  012767  036104  163132          MOV     #RBUFF,DATAD    ;SET BUFFER ADDRESS
2129  015626  012767  035472  163122          MOV     #WBUFF,WTAD     ;SET POINTER TO WRITE BUFFER
2130  015634  005767  163146                  TST     RDRVF
2131  015640  001411                          BEQ     WAM3A
2132  015642  062767  000376  163110          ADD     #376,DATAD
2133  015650  062767  000377  163100          ADD     #377,WTAD
2134  015656  012767  000076  163066          MOV     #76,FUN         ;SET READ REVERSE CODE
2135  015664  032767  000020  163104  WAM3A:  BIT     #20,UDES
2136  015672  001403                          BEQ     WAM3B
2137  015674  016767  163110  163054          MOV     RCDP,WTAD
2138  015702  004767  000004          WAM3B:  JSP     PC,SETUP        ;GO SET UP
2139  015706  000167  000332                  JMP     EXEC            ;GO EXECUTE
2140
```

```
2141                                                    ;REGISTER SETUP ROUTINE*****
2142
2143  015712  005767  163022          SETUP:  TST     STFLG           ;SEE IF SINGLE TEST
2144  015716  001403                          BEQ     SET0            ;IF NOT; BR
2145  015720  005767  163054                  TST     WPGFL           ;SEE IF HAVE SELECTED PATTERN
2146  015724  001002                          BNE     SET1            ;IF SO; BR
2147  015726  004767  177262          SET0:   JSR     PC,DSUP         ;GO DO DATA SETUP
2148  015732  004767  005544          SET1:   JSR     PC,INIT1        ;GO INIT SELECT DRIVE, SLAVE
2149  015736  004767  000656                  JSR     PC,BOTT         ;GO ASSURE NOT AT BOT
2150  015742  012777  177400  162546          MOV     #-400,@FC       ;SET FC=WCX2
2151  015750  032767  000020  163020          BIT     #20,UDES        ;SEE IF CORE DUMP
2152  015756  001403                          BEQ     SET2            ;IF NOT; BR
2153  015760  012777  177000  162530          MOV     #-1000,@FC      ;SET FC=WCX4
2154  015766  012777  177600  162516  SET2:   MOV     #-200,@WC       ;SET WC
2155  015774  016777  162760  162512          MOV     DATAD,@BA       ;SET BUS ADDRESS
2156  016002  032777  010000  162510          BIT     #10000,@CS      ;ASSURE DRIVE THERE
2157  016010  001420                          BEQ     SP1             ;IF SO; BR
2158  016012  032777  020000  162550          BIT     #20000,@SWR     ;SEE IF PRINT ERRORS
2159  016020  001004                          BNE     SP01            ;IF NOT; BR
2160  016022  012704  034603                  MOV     #WMSG4,R4
2161  016026  004767  006012                  JSR     PC,TTOUT        ;PRINT NON-EXISTANT DRIVE
2162  016032  032777  100000  162530  SP01:   BIT     #100000,@SWR    ;SEE IF HALT ON ERROR
2163  016040  001402                          BEQ     SP0             ;IF NOT; BR
2164  016042  004767  007114                  JSR     PC,STOP
2165  016046  000167  177660          SP0:    JMP     SET1            ;RESETUP
2166  016052  022767  000014  162670  SP1:    CMP     #14,WAM         ;SEE IF WAM ?
2167  016060  001026                          BNE     SP1B            ;IF NOT; BR
2168  016062  117767  162670  162664          MOVB    @WTAD,DATC      ;GET FIRST CHAR
2169  016070  042767  177400  162656          BIC     #177400,DATC
2170  016076  000367  162652                  SWAB    DATC
2171  016102  005767  162700                  TST     RDRVF           ;SEE IF READ REVERSE
2172  016106  001403                          BEQ     SP1A            ;IF NOT; BR
2173  016110  005367  162642                  DEC     WTAD            ;DECREMENT POINTER
2174  016114  000410                          BR      SP1B
2175  016116  005267  162634          SP1A:   INC     WTAD            ;BUMP POINTER
2176  016122  032767  000020  162646          BIT     #20,UDES        ;SEE IF CORE DUMP
2177  016130  001402                          BEQ     SP1B            ;IF NOT; BR
2178  016132  005267  162620                  INC     WTAD            ;BUMP POINTER AGAIN
2179  016136  056777  162634  162376  SP1B:   BIS     UDES,@TC        ;SET UNIT DESCRIPTION (DEN,PAR,FMT)
2180  016144  052777  000001  162362          BIS     #1,@MR          ;SET MAINT MODE
2181  016152  056777  162572  162354          BIS     WAM,@MR         ;SET WAM
2182  016160  056777  162570  162346          BIS     DATC,@MR        ;SET DATA
2183  016166  016777  162560  162314          MOV     FUN,@C1         ;SET FUNCTION
2184  016174  032777  040000  162320          BIT     #40000,@DS      ;ASSURE NO ERROR
2185  016202  001001                          BNE     SP3             ;IF NOT; BR
2186  016204  000207                          RTS     PC              ;RETURN
2187  016206  032777  020000  162354  SP3:    BIT     #20000,@SWR     ;SEE IF PRINT ERRORS
2188  016214  001004                          BNE     SP4             ;IF NOT; BR
2189  016216  012704  034544                  MOV     #WMSG2,R4
2190  016222  004767  005616                  JSR     PC,TTOUT        ;PRINT SETUP ERROR
2191  016226  032777  100000  162334  SP4:    BIT     #100000,@SWR    ;SEE IF HALT ON ERROR
2192  016234  001402                          BEQ     SP5             ;IF NOT; BR
2193  016236  004767  006720                  JSR     PC,STOP
2194  016242  000207                  SP5:    RTS     PC              ;RETURN
```

```
2195                                                ;EXECUTE WAM ROUTINE*****
2196
2197  016244  032777  000040  162262  EXEC:  BIT     #40,@MR
2198  016252  001403                        BEQ     EX0             ;ASSURE MAINT CLOCK IS ZERO
2199  016254  042777  000040  162252        BIC     #40,@MR         ;IF NOT: CLEAR IT
2200  016262  022767  000010  162460  EX0:   CMP     #10,WAM         ;SEE IF WAM 1 OR 2 OR 3
2201  016270  003402                        BLE     EX1             ;IF SO: BR
2202  016272  000167  000364                JMP     EXW2            ;GO DO WAM 0
2203  016276  052777  000001  162284  EX1:   BIS     #1,@PC1         ;SET GO BIT
2204  016304  005000                        CLR     R0
2205  016306  012701  000002                MOV     #2,R1           ;SET DELAY
2206  016312  032777  100000  162222  EX1A:  BIT     #100000,@TC     ;SEE IF ALPHA
2207  016320  001404                        BEQ     EX2             ;IF SO: BR
2208  016322  005300                        DEC     R0
2209  016324  001372                        BNE     EX1A            ;AWAIT ALPHA
2210  016326  005301                        DEC     R1
2211  016330  001370                        BNE     EX1A
2212  016332  005077  162230          EX2:  CLR     @PSW
2213  016336  012701  000400                MOV     #400,R1         ;SET NUMBER OF CLKS
2214  016342  032767  000020  162426        BIT     #20,UDES        ;SEE IF CORE DUMP
2215  016350  001402                        BEQ     EX3             ;IF NOT: BR
2216  016352  012701  001000                MOV     #1000,R1        ;SET CLOCKS LWCX4
2217  016356  022767  000014  162364  EX3:  CMP     #14,WAM         ;SEE IF WAM 3
2218  016364  001413                        BEQ     EX5A            ;IF SO: BR
2219  016366  032767  002000  162402        BIT     #2000,UDES      ;SEE IF PE
2220  016374  001405                        BEQ     EX5             ;IF NOT PE: BR
2221  016376  000241                        CLC
2222  016400  006101                        ROL     R1
2223  016402  062701  000246                ADD     #246,R1         ;SET TO ALLOW FOR PRE/POSTAMBLE
2224  016406  000402                        BR      EX5A
2225  016410  062701  000010          EX5:  ADD     #10,R1          ;ADD CLOCKS FOR CRC AND LRC
2226  016414  022767  000014  162326  EX5A:  CMP     #14,WAM         ;SEE IF WAM 3
2227  016422  001044                        BNE     EX5C            ;IF NOT: BR
2228  016424  117700  162326                MOVB    @WTAD,R0
2229  016430  042700  177400                BIC     #177400,R0
2230  016434  005767  162346                TST     RDRVF           ;SEE IF REVERSE
2231  016440  001403                        BEQ     EX5A1           ;IF NOT: BR
2232  016442  005367  162310                DEC     WTAD            ;DEC POINTER
2233  016446  000416                        BR      EX5B
2234  016450  005267  162302          EX5A1: INC     WTAD
2235  016454  032767  000020  162314        BIT     #20,UDES        ;SEE IF CORE DUMP
2236  016462  001410                        BEQ     EX5B            ;IF NOT: BR
2237  016464  005267  162266                INC     WTAD            ;BUMP POINTER
2238  016470  005777  162262                TST     @WTAD           ;SEE IF END
2239  016474  001003                        BNE     EX5B            ;IF NOT: BR
2240  016476  162767  000010  162252        SUB     #10,WTAD        ;RESTORE POINTER
2241  016504  052777  000040  162022  EX5B:  BIS     #40,@MR         ;CLOCK UP
2242  016512  017702  162016                MOV     @MR,R2          ;READ MR
2243  016516  042702  177400                BIC     #177400,R2      ;MASK OUT DATA
2244  016522  000300                        SWAB    R0              ;POSITION DATA
2245  016524  050002                        BIS     R0,R2           ;LOAD NEW DATA
2246  016526  010277  162002                MOV     R2,@MR          ;CLOCK DOWN AND LOAD NEW DATA
2247  016532  000426                        BR      EX5D
2248  016534  052777  000040  161772  EX5C:  BIS     #40,@MR         ;CLOCK UP
2249  016542  042777  000040  161764        BIC     #40,@MR         ;CLOCK DOWN
2250  016550  017700  161760                MOV     @MR,R0          ;GET MR
```

```
2251  016554  000300                          SWAB    R0
2252  016556  032767  000010  162212          BIT     #10,UDES        ;SEE IF EVEN PAR
2253  016564  001405                          BEQ     EX5C0           ;IF NOT: BR
2254  016566  010077  162170                  MOV     R0,@RDAD
2255  016572  005267  162164                  INC     RDAD
2256  016576  000402                          BR      EX5C1
2257  016600  110077  162156          EX5C0:  MOVB    R0,@RDAD        ;PUT CHAR IN CORE
2258  016604  005267  162152          EX5C1:  INC     RDAD
2259  016610  005301                  EX5D:   DEC     R1              ;SEE IF DONE CLKS
2260  016612  001300                          BNE     EX5A            ;IF NOT: BR
2261  016614  000167  002762                  JMP     EORP            ;GO DO EOR
2262
2263                                          ;ASSURE NOT AT BOT FOR WRAP TESTS**********
2264
2265  016620  032777  000002  161674  BOTT:   BIT     #2,@DS          ;SEE IF BOT
2266  016626  001414                          BEQ     BOTTX           ;IF NOT: BR
2267  016630  052777  001700  161704          BIS     #1700,@TC       ;SET NRZ
2268  016636  012777  000025  161644          MOV     #25,@C1         ;DO ERASE
2269  016644  032777  000200  161650  BOTTA:  BIT     #200,@DS
2270  016652  001774                          BEQ     BOTTA           ;AWAIT DRY
2271  016654  004767  004622                  JSR     PC,INIT1        ;INIT
2272  016660  000207                  BOTTX:  RTS     PC              ;RETURN
2273
```

```
 2274                                                ;EXECUTE WAM 8******
 2275
 2276  016662  012767  017034  162000  EXW2:   MOV     #EXW2H,RTRN     ;SET INTERRUPT RETURN ADDRESS
 2277  016670  012701  000200                  MOV     #200,R1         ;SET NUMBER OF CLOCKS = FC/2
 2278  016674  032767  002000  162074          BIT     #2000,UDES      ;SEE IF PE
 2279  016702  001402                          BEQ     EXW2A           ;IF NOT; BR
 2280  016704  012701  000100                  MOV     #100,R1         ;ELSE SET CLKS = FC/4
 2281  016710  016702  162046          EXW2A:  MOV     RDAD,R2         ;SET BUFFER ADDRESS
 2282  016714  012777  000161  161566          MOV     #161,@C1        ;SET WRITE+GO
 2283  016722  005077  161640                  CLR     @PSW            ;ALLOW INTERRUPT
 2284  016726  032777  000040  161600  EXW2B:  BIT     #40,@MR
 2285  016734  001774                          BEQ     EXW2B           ;AWAIT CLOCK UP
 2286  016736  017722  161572                  MOV     @MR,(R2)+       ;GET DATA
 2287  016742  032777  000040  161564  EXW2C:  BIT     #40,@MR
 2288  016750  001374                          BNE     EXW2C           ;AWAIT CLOCK DOWN
 2289  016752  017722  161556                  MOV     @MR,(R2)+       ;GET DATA
 2290  016756  005301                          DEC     R1              ;SEE IF DONE ALL
 2291  016760  001362                          BNE     EXW2B           ;IF NOT; BR
 2292  016762  012701  000003          EXW2E:  MOV     #3,R1
 2293  016766  005000                          CLR     R0              ;SET DELAY
 2294  016770  005300                  EXW2F:  DEC     R0
 2295  016772  001376                          BNE     EXW2F
 2296  016774  005301                          DEC     R1
 2297  016776  001374                          BNE     EXW2F           ;DELAY
 2298  017000  032777  020000  161562          BIT     #20000,@SWR     ;SEE IF ERROR PRINT
 2299  017006  001004                          BNE     EXW2G           ;IF NOT; BR
 2300  017010  012704  035006                  MOV     #WMSG24,R4
 2301  017014  004767  005024                  JSR     PC,TTOUT        ;PRINT NO INTERUPT
 2302  017020  032777  100000  161542  EXW2G:  BIT     #100000,@SWR    ;SEE IF HALT ON ERROR
 2303  017026  001402                          BEQ     EXW2H           ;IF NOT; BR
 2304  017030  004767  006126                  JSR     PC,STOP
 2305  017034  012701  036104          EXW2H:  MOV     #RBUFF,R1       ;GET START OF READ BUFFER
 2306  017040  012700  000400                  MOV     #400,R0         ;SET SIZE
 2307  017044  010102                          MOV     R1,R2
 2308  017046  012203                  EXW2J:  MOV     (R2)+,R3
 2309  017050  000303                          SWAB    R3
 2310  017052  032767  000010  161716          BIT     #10,UDES        ;SEE IF EVEN PAR
 2311  017060  001402                          BEQ     EXW2J0          ;IF NOT; BR
 2312  017062  010321                          MOV     R3,(R1)+        ;SAVE PAR + DATA
 2313  017064  000401                          BR      EXW2J1
 2314  017066  110321                  EXW2J0: MOVB    R3,(R1)+        ;ASSEMBLE DATA IN BYTES
 2315  017070  005300                  EXW2J1: DEC     R0
 2316  017072  001365                          BNE     EXW2J           ;CONTINUE FOR ALL
 2317  017074  032777  000200  161466          BIT     #200,@SWR       ;SEE IF STATUS CHECK
 2318  017102  001002                          BNE     EXW2K           ;IF NOT; BR
 2319  017104  004767  000016                  JSR     PC,WSTCK        ;ELSE GO CHECK STATUS
 2320  017110  032777  000400  161452  EXW2K:  BIT     #400,@SWR       ;SEE IF DATA CHECK
 2321  017116  001002                          BNE     EXW2X           ;IF NOT; BR
 2322  017120  004767  000272                  JSR     PC,DCHK         ;ELSE GO CHECK DATA
 2323  017124  000207                  EXW2X:  RTS     PC              ;EXIT
```

```
                                        ;WRAP AROUND STATUS CHECK ROUTINE*********
017126  005067  161640        WSTCK:  CLR     SERFL           ;CLEAR ERROR FLAG
017132  005067  161462                CLR     HDRFL           ;CLEAR HEADER FLAG
017136  012767  034630  161526        MOV     #WMSG6,ERADD    ;SET CODE=CS1
017144  017702  161340                MOV     @C1,R2          ;GET RCVD CS1
017150  016705  161644                MOV     WCS1,R5         ;GET EXPT CS1
017154  004767  000114                JSR     PC,WSTG         ;GO CHK
017160  012767  034655  161504        MOV     #WMSG6D,ERADD   ;SET CODE=CS2
017166  017702  161326                MOV     @CS,R2          ;SET RCVD CS2
017172  016705  161624                MOV     WCS2,R5         ;GET EXPT CS2
017176  056705  161422                BIS     DRVN,R5         ;SET DRIVE NUMBER IN EXPT CS2
017202  004767  000066                JSR     PC,WSTG         ;GO CHK
017206  012767  034663  161456        MOV     #WMSG6E,ERADD   ;SET CODE=DS
017214  017702  161302                MOV     @DS,R2          ;SET RCVD DS
017220  016705  161600                MOV     WDS,R5          ;GET EXPT DS
017224  004767  000044                JSR     PC,WSTG         ;GO CHK
017230  012767  034670  161434        MOV     #WMSG6F,ERADD   ;SET CODE=ER
017236  017702  161262                MOV     @ER,R2          ;GET RCVD ER
017242  016705  161560                MOV     WER,R5          ;GET EXPT ER
017246  004767  000022                JSR     PC,WSTG         ;GO CHK
017252  005767  161514                TST     SERFL           ;SEE IF ANY ERRORS
017256  001456                        BEQ     WSTX            ;IF NOT: BR
017260  005777  161304                TST     @SWR            ;SEE IF HALT ON ERROR
017264  100053                        BPL     WSTX            ;IF NOT: BR
017266  004767  005670                JSR     PC,$TOP
017272  000450                        BR      WSTX            ;CONTINUE
017274  020205                WSTG:   CMP     R2,R5           ;SEE IF EXPT=RCVD
017276  001446                        BEQ     WSTX            ;IF SO: BR
017300  005267  161466                INC     SERFL           ;SET ERROR FLAG
017304  032777  020000  161256        BIT     #20000,@SWR     ;SEE IF PRINT ERRORS
017312  001040                        BNE     WSTX            ;IF NOT: BR
017314  005767  161300                TST     HDRFL           ;SEE IF DONE HEADER
017320  001010                        BNE     WSTG0           ;IF SO: BR
017322  016704  161274                MOV     EMADDR,R4
017326  004767  004512                JSR     PC,TTOUT        ;PRINT TEST HEADER
017332  012704  034772                MOV     #WMSG23,R4
017336  004767  004502                JSR     PC,TTOUT        ;PRINT STATUS TAG
017342  012767  000001  161250  WSTG0: MOV     #1,HDRFL        ;SET HEADER FLAG
017350  016704  161316                MOV     ERADD,R4
017354  004767  004464                JSR     PC,TTOUT        ;PRINT CODE
017360  012704  026022                MOV     #MSG12,R4
017364  004767  004454                JSR     PC,TTOUT        ;PRINT EXPT TAG
017370  010503                        MOV     R5,R3
017372  004767  004610                JSR     PC,OCTP         ;PRINT EXPT STATUS
017376  012704  026031                MOV     #MSG13,R4
017402  004767  004436                JSR     PC,TTOUT        ;PRINT RCVD TAG
017406  010203                        MOV     R2,R3
017410  004767  004572                JSR     PC,OCTP         ;PRINT RCVD STATUS
017414  000207                WSTX:   RTS     PC              ;RETURN
```

```
 2376
 2377                                        ;DATA CHECK ROUTINE*****
 2378
 2379  017416  005067  161176          DCHK:  CLR     HDRFL            ;CLEAR HEADER FLAG
 2380  017422  005067  161340                 CLR     DERFL            ;CLEAR DATA ERROR FLAG
 2381  017426  005067  161342                 CLR     CRCNT            ;CLEAR CHAR CNTR
 2382  017432  032767  000010  161336         BIT     #10,UDES         ;SEE IF EVEN PARITY
 2383  017440  001402                         BEQ     DCHKA0           ;IF NOT: BR
 2384  017442  000167  000602                 JMP     DCHKE            ;ELSE GO CHECK EVEN
 2385  017446  022767  000010  161274  DCHKA0: CMP     #10,WAM          ;SEE IF WAM 1
 2386  017454  001006                         BNE     DCHKA            ;IF NOT: BR
 2387  017456  032767  002000  161312         BIT     #2000,UDES       ;SEE IF PE
 2388  017464  001402                         BEQ     DCHKA            ;IF NOT: BR
 2389  017466  000167  001202                 JMP     PRCHK            ;GO CHK DATA
 2390  017472  012700  177400          DCHKA: MOV     #-400,R0         ;SET NUMBER OF CHARACTERS
 2391  017476  022767  000012  161244         CMP     #12,WAM
 2392  017504  001006                         BNE     DCHKA1           ;IF NOT WRAP 2: BR
 2393  017506  032767  000020  161262         BIT     #20,UDES
 2394  017514  001402                         BEQ     DCHKA1           ;IF NOT CORE DUMP: BR
 2395  017516  012700  177000                 MOV     #-1000,R0
 2396  017522  022767  000006  161220  DCHKA1: CMP     #6,WAM
 2397  017530  001007                         BNE     DCHKA2
 2398  017532  032767  002000  161236         BIT     #2000,UDES       ;SEE IF PE MODE
 2399  017540  001430                         BEQ     DCHKB            ;IF NOT: BR
 2400  017542  012700  177600                 MOV     #-200,R0         ;SET CHAR CNTR TO FC/2 FOR PE
 2401  017546  000425                         BR      DCHKB
 2402  017550  022767  000012  161172  DCHKA2: CMP     #12,WAM          ;SEE IF WRAP 2
 2403  017556  001021                         BNE     DCHKB            ;IF NOT: BR
 2404  017560  032767  002000  161210         BIT     #2000,UDES       ;SEE IF PE
 2405  017566  001415                         BEQ     DCHKB            ;IF NOT: BR
 2406  017570  012700  177653                 MOV     #-125,R0         ;POINT TO START OF DATA
 2407  017574  012767  000001  161162         MOV     #1,W2FLG         ;SET WRAP 2 FLAG
 2408  017602  004767  000014                 JSR     PC,DCHKB         ;GO CHECK DATA
 2409  017606  004767  001362                 JSR     PC,W1DCHK        ;GO CHECK WRAP 1 DATA
 2410  017612  005067  161146                 CLR     W2FLG
 2411  017616  000167  000362                 JMP     DCHKX
 2412  017622  005067  161140          DCHKB: CLR     DERFL
 2413  017626  012701  035472                 MOV     #WBUFF,R1        ;SET GOOD POINTER
 2414  017632  012702  036104                 MOV     #RBUFF,R2        ;SET READ POINTER
 2415  017636  032767  000020  161132         BIT     #20,UDES         ;SEE IF CORE DUMP
 2416  017644  001416                         BEQ     DCHK0            ;IF NOT: BR
 2417  017646  022767  000012  161074         CMP     #12,WAM          ;SEE IF WAM 2
 2418  017654  001011                         BNE     DCHKD            ;IF NOT: BR
 2419  017656  005767  161120                 TST     PATRN            ;SEE IF PATTERN 0
 2420  017662  001003                         BNE     DCHKC            ;IF NOT: BR
 2421  017664  012701  001054                 MOV     #WCDP0,R1        ;SET CORE DUMP PATTERN 0
 2422  017670  000404                         BR      DCHK0            ;GO CHECK DATA
 2423  017672  012701  001042          DCHKC: MOV     #WCDP2,R1        ;SET CORE DUMP WRITE PATTERN 2
 2424  017676  000401                         BR      DCHK0            ;GO CHECK DATA
 2425  017700  000240                  DCHKD: NOP
 2426  017702  121112                  DCHK0: CMPB    (R1),(R2)        ;SEE IF DATA OK
 2427  017704  001466                         BEQ     DCHK2            ;IF SO: BR
 2428  017706  032777  020000  160654         BIT     #20000,@SWR      ;SEE IF PRINT ERRORS
 2429  017714  001062                         BNE     DCHK2            ;IF NOT: BR
 2430  017716  005767  160676                 TST     HDRFL            ;SEE IF DONE HEADER
 2431  017722  001004                         BNE     DCHK1            ;IF SO: BR
```

```
 2432  017724  016704  160672                    MOV     EMADDR,R4
 2433  017730  004767  004110                    JSR     PC,TTOUT        ;PRINT HEADER
 2434  017734  005767  161026          DCHK1:    TST     DERFL           ;SEE IF FIRST ERROR
 2435  017740  001014                            BNE     DCHK1A          ;IF NOT: BR
 2436  017742  012704  034740                    MOV     #WMSG16,R4
 2437  017746  004767  004072                    JSR     PC,TTOUT        ;PRINT DATA ERROR TAG
 2438  017752  012704  035165                    MOV     #WMSG32,R4
 2439  017756  004767  004062                    JSR     PC,TTOUT        ;PRINT PATRN TAG
 2440  017762  016703  161014                    MOV     PATRN,R3
 2441  017766  004767  004214                    JSR     PC,OCTP         ;PRINT PATTERN NUMBER
 2442  017772  012767  000001  160620  DCHK1A:   MOV     #1,HDRFL        ;SET HEADER FLAG
 2443  020000  012767  000001  160760            MOV     #1,DERFL        ;SET DATA ERROR FLAG
 2444  020006  012704  034764                    MOV     #WMSG21,R4
 2445  020012  004767  004026                    JSR     PC,TTOUT        ;PRINT CHARACTER NUMBER TAG
 2446  020016  016703  160752                    MOV     CRCNT,R3
 2447  020022  004767  004160                    JSR     PC,OCTP         ;PRINT CHARACTER NUMBER
 2448  020026  012704  034752                    MOV     #WMSG17,R4
 2449  020032  004767  004006                    JSR     PC,TTOUT        ;PRINT GOOD TAG
 2450  020036  111103                            MOVB    (R1),R3
 2451  020040  004767  004706                    JSR     PC,DOUT         ;PRINT GOOD DATA
 2452  020044  012704  034757                    MOV     #WMSG20,R4
 2453  020050  004767  003770                    JSR     PC,TTOUT        ;PRINT BAD TAG
 2454  020054  111203                            MOVB    (R2),R3
 2455  020056  004767  004670                    JSR     PC,DOUT         ;PRINT BAD DATA
 2456  020062  005767  160676          DCHK2:    TST     W2FLG
 2457  020066  001020                            BNE     DCHK2B
 2458  020070  005201                            INC     R1              ;BUMP POINTER
 2459  020072  032767  000020  160676            BIT     #20,UDES        ;SEE IF CORE DUMP
 2460  020100  001413                            BEQ     DCHK2B          ;IF NOT: BR
 2461  020102  022767  000012  160640            CMP     #12,WAM         ;SEE IF WAM 2
 2462  020110  001006                            BNE     DCHK2A          ;IF NOT: BR
 2463  020112  005201                            INC     R1              ;BUMP POINTER
 2464  020114  005711                            TST     (R1)            ;SEE IF END OF PATTERN
 2465  020116  001004                            BNE     DCHK2B          ;IF NOT: BR
 2466  020120  162701  000010                    SUB     #10,R1          ;RESET POINTER TO START OF PATTERN
 2467  020124  000401                            BR      DCHK2B          ;CONTINUE CHECK
 2468  020126  000240          DCHK2A:   NOP
 2469  020130  005202          DCHK2B:   INC     R2
 2470  020132  022767  000006  160610            CMP     #6,WAM          ;SEE IF WAM 6
 2471  020140  001005                            BNE     DCHK3
 2472  020142  032767  002000  160626            BIT     #2000,UDES      ;SEE IF PE
 2473  020150  001401                            BEQ     DCHK3           ;IF NOT PE: BR
 2474  020152  005201                            INC     R1              ;BUMP WRITE DATA ADDRESS
 2475  020154  005267  160614          DCHK3:    INC     CRCNT           ;BUMP CHAR CNTR
 2476  020160  032777  000400  160402            BIT     #400,@SWR       ;SEE IF CONT DATA CHK
 2477  020166  001006                            BNE     DCHKX           ;IF NOT: BR
 2478  020170  005200                            INC     R0              ;SEE IF DONE
 2479  020172  001243                            BNE     DCHK0           ;IF NOT: BR
 2480  020174  005767  160564                    TST     W2FLG
 2481  020200  001401                            BEQ     DCHKX
 2482  020202  000207                            RTS     PC
 2483  020204  032777  100000  160356  DCHKX:    BIT     #100000,@SWR    ;SEE IF HALT ON ERROR
 2484  020212  001405                            BEQ     DCHKX1          ;IF NOT: BR
 2485  020214  005767  160546                    TST     DERFL           ;SEE IF DATA ERROR OCCURED
 2486  020220  001402                            BEQ     DCHKX1          ;IF NOT: BR
 2487  020222  004767  004734                    JSR     PC,STOP
```

```
 2488  020226  005067  160542          DCHKX1: CLR     CRCNT           ;CLEAR CHAR CNTR
 2489  020232  005067  160362                  CLR     HDRFL           ;CLEAR HEADER FLAG
 2490  020236  005067  160524                  CLR     DERFL           ;CLEAR DATA ERROR FLAG
 2491  020242  005067  160522                  CLR     PREFL           ;CLEAR PREAMBLE FLAG
 2492  020246  000207                          RTS     PC              ;RETURN
```

```
 2493
 2494                                        ;EVEN PARITY DATA CHECK*****
 2495
 2496  020250  012700  177400          DCHKE:  MOV     #-400,R0        ;SET NUMBER OF CHARACTERS
 2497  020254  012701  035472                  MOV     #WBUFF,R1       ;R1=START OF WRITE BUFFER
 2498  020260  012702  036104                  MOV     #RBUFF,R2       ;R2=START OF READ BUFFER
 2499  020264  111105                  DCKE0:  MOVB    (R1),R5         ;GET EXPT DATA
 2500  020266  005003                          CLR     R3
 2501  020270  012704  000010                  MOV     #10,R4          ;SET NUMBER OF BITS
 2502  020274  032705  000001          DCKE1:  BIT     #1,R5           ;SEE IF ONE BIT
 2503  020300  001401                          BEQ     DCKE2           ;IF NOT; BR
 2504  020302  005203                          INC     R3              ;COUNT ONE BITS FOR PARITY CHECK
 2505  020304  005304                  DCKE2:  DEC     R4              ;SEE IF DONE
 2506  020306  001402                          BEQ     DCKE3           ;IF SO; BR
 2507  020310  006005                          ROR     R5              ;POINT TO NEXT BIT
 2508  020312  000770                          BR      DCKE1
 2509  020314  111105                  DCKE3:  MOVB    (R1),R5         ;GET EXPT DATA
 2510  020316  042705  177400                  BIC     #177400,R5      ;MASK DATA FIELD
 2511  020322  005703                          TST     R3
 2512  020324  001003                          BNE     DCKE4           ;IF NO ONE BITS SET; BR
 2513  020326  012705  100020                  MOV     #100020,R5
 2514  020332  000405                          BR      DCKE5
 2515  020334  032703  000001          DCKE4:  BIT     #1,R3           ;SEE IF ODD NUMBER OF ONE BITS
 2516  020340  001402                          BEQ     DCKE5           ;IF NOT; BR
 2517  020342  052705  100000                  BIS     #100000,R5      ;SET EVEN PARITY BIT=1
 2518  020346  042712  077400          DCKE5:  BIC     #77400,(R2)     ;MASK DATA FIELD
 2519  020352  020512                          CMP     R5,(R2)         ;SEE IF DATA + PARITY GOOD
 2520  020354  001474                          BEQ     DCKE10          ;IF SO; BR
 2521  020356  032777  020000  160204          BIT     #20000,@SWR     ;SEE IF ERROR PRINT
 2522  020364  001070                          BNE     DCKE10          ;IF NOT; BR
 2523  020366  005767  160226                  TST     HDRFL           ;SEE IF DONE HEADER
 2524  020372  001004                          BNE     DCKE6           ;IF SO; BR
 2525  020374  016704  160222                  MOV     EMADDR,R4
 2526  020400  004767  003440                  JSR     PC,TTOUT        ;PRINT HEADER
 2527  020404  005767  160356          DCKE6:  TST     DERFL           ;SEE IF FIRST BAD CHAR
 2528  020410  001014                          BNE     DCKE7           ;IF NOT; BR
 2529  020412  012704  034740                  MOV     #WMSG16,R4
 2530  020416  004767  003422                  JSR     PC,TTOUT        ;PRINT BAD DATA TAG
 2531  020422  012704  035165                  MOV     #WMSG32,R4
 2532  020426  004767  003412                  JSR     PC,TTOUT        ;PRINT PATTERN TAG
 2533  020432  016703  160344                  MOV     PATRN,R3
 2534  020436  004767  003544                  JSR     PC,OCTP         ;PRINT PATTERN NUMBER
 2535  020442  012767  000001  160316  DCKE7:  MOV     #1,DERFL        ;SET DATA ERROR FLAG
 2536  020450  012767  000001  160142          MOV     #1,HDRFL        ;SET HEADER FLAG
 2537  020456  012704  034764                  MOV     #WMSG21,R4
 2538  020462  004767  003356                  JSR     PC,TTOUT        ;PRINT CHAR NUMBER TAG
 2539  020466  016703  160302                  MOV     CRCNT,R3
 2540  020472  004767  003510                  JSR     PC,OCTP         ;ORINT CHAR NUMBER
 2541  020476  012704  034752                  MOV     #WMSG17,R4
 2542  020502  004767  003336                  JSR     PC,TTOUT        ;PRINT GOOD DATA TAG
 2543  020506  110503                          MOVB    R5,R3
 2544  020510  004767  004236                  JSR     PC,DOUT         ;PRINT EXPT DATA
 2545  020514  010503                          MOV     R5,R3
 2546  020516  004767  000064                  JSR     PC,DCKEP        ;GO PRINT PARITY BIT
 2547  020522  012704  034757                  MOV     #WMSG20,R4
 2548  020526  004767  003312                  JSR     PC,TTOUT        ;PRINT BAD T
```

```
 2549  020532  111203                          MOVB    (R2),R3
 2550  020534  004767  004212                  JSR     PC,DOUT         ;PRINT BAD DATA
 2551  020540  011203                          MOV     (R2),R3
 2552  020542  004767  000040                  JSR     PC,DCKEP        ;GO PRINT PARITY BIT
 2553  020546  005201                  DCKE10: INC     R1
 2554  020550  005722                          TST     (R2)+           ;BUMP POINTERS
 2555  020552  005267  160216                  INC     CRCNT           ;BUMP CHAR CNTR
 2556  020556  032777  000400  160004          BIT     #400,@#WR       ;SEE IF CONTINUE DATA CHECK
 2557  020564  001402                          BEQ     DCKE11          ;IF SO: BR
 2558  020566  000167  177412                  JMP     DCHKX           ;GO TO END OF DATA CHECK
 2559  020572  005200                  DCKE11: INC     R0              ;SEE IF DONE
 2560  020574  001402                          BEQ     DCKE12          ;IF SO: BR
 2561  020576  000167  177462                  JMP     DCKE0           ;ELSE CONTINUE
 2562  020602  000167  177376          DCKE12: JMP     DCHKX           ;GO TO END OF DATA CHECK
 2563  020606  012767  000240  160000  DCKEP:  MOV     #240,TOB
 2564  020614  004767  003340                  JSR     PC,TOG          ;SPACE
 2565  020620  012767  000260  157766          MOV     #260,TOB        ;SET PAR=0
 2566  020626  005703                          TST     R3              ;SEE IF PARITY REALLY=0
 2567  020630  100002                          BPL     DCKEP0          ;IF SO: BR
 2568  020632  005267  157756                  INC     TOB             ;ELSE SET TO 1
 2569  020636  004767  003316          DCKEP0: JSR     PC,TOG          ;PRINT PARITY BIT
 2570  020642  000207                          RTS     PC              ;RETURN
 2571
```

```
 2572
 2573                                        ;PREAMPLE/POSTAMPLE CHECK*******
 2574
 2575  020644  012700  000051        PSCHK:  MOV     #51,R0          ;SET SIZE OF POSTAMBLE
 2576  020650  012701  035350                MOV     #POST,R1        ;SET POINTER TO POSTAMBLE
 2577  020654  005067  157740                CLR     HDRFL           ;CLEAR HEADER FLAG
 2578  020660  005067  160110                CLR     CRCNT           ;CLEAR CHAR CNTR
 2579  020664  005067  160076                CLR     DERFL           ;CLEAR DATA ERROR FLAG
 2580  020670  000167  000016                JMP     PD0             ;GO CHECK POSTAMPLE
 2581
 2582  020674  012700  000051        PRCHK:  MOV     #51,R0          ;SET SIZE OF PREAMBLE
 2583  020700  012701  035226                MOV     #PRE,R1         ;SET POINTER TO PREAMBLE
 2584  020704  012702  036104                MOV     #RBUFF,R2       ;SET POINTER TO START OF READ BUFFER
 2585  020710  022122                        CMP     (R1)+,(R2)+     ;BUMP ADDRESS POINTERS
 2586  020712  121112        PD0:    CMPB    (R1),(R2)       ;CHECK DATA
 2587  020714  001004                BNE     PD1             ;IF NOT GOOD: BR
 2588  020716  126162  000001  000001  CMPB    1(R1),1(R2)     ;COMPARE COMPLIMENT BYTE
 2589  020724  001477                BEQ     PD5             ;IF GOOD: BR
 2590  020726  032777  020000  157634  PD1:    BIT     #20000,@SWR    ;SEE IF PRINT INHIBIT
 2591  020734  001073                BNE     PD5             ;IF SO: BR
 2592  020736  005767  157656                TST     HDRFL           ;SEE IF DONE HEADER
 2593  020742  001020                BNE     PD4             ;IF SO: BR
 2594  020744  016704  157652                MOV     EMADDR,R4
 2595  020750  004767  003070                JSR     PC,TTOUT        ;PRINT TEST HEADER
 2596  020754  005767  160010                TST     PREFL           ;SEE IF PREAMBLE CHECK
 2597  020760  001403                BEQ     PD2             ;IF NOT: BR
 2598  020762  012704  035111                MOV     #WMSG29,R4      ;SET POSTAMBLE HEADER
 2599  020766  000402                BR      PD3
 2600  020770  012704  035073        PD2:    MOV     #WMSG28,R4      ;SET PREAMBLE HEADER
 2601  020774  004767  003044        PD3:    JSR     PC,TTOUT        ;PRINT HEADER
 2602  021000  005267  157614                INC     HDRFL
 2603  021004  012704  034764        PD4:    MOV     #WMSG21,R4
 2604  021010  004767  003030                JSR     PC,TTOUT        ;PRINT CHAR NUMBER TAG
 2605  021014  016703  157754                MOV     CRCNT,R3
 2606  021020  004767  003162                JSR     PC,OCTP         ;PRINT CHAR NUMBER
 2607  021024  012704  034752                MOV     #WMSG17,R4
 2608  021030  004767  003010                JSR     PC,TTOUT        ;PRINT GOOD TAG
 2609  021034  116103  000001                MOVB    1(R1),R3
 2610  021040  004767  003706                JSR     PC,DOUT         ;PRINT GOOD CHAR
 2611  021044  012767  000240  157542        MOV     #240,TOB
 2612  021052  004767  003102                JSR     PC,TOG
 2613  021056  111103                MOVB    (R1),R3
 2614  021060  004767  003666                JSR     PC,DOUT         ;PRINT COMPLIMENT
 2615  021064  012704  034757                MOV     #WMSG28,R4
 2616  021070  004767  002750                JSR     PC,TTOUT        ;PRINT BAD TAG
 2617  021074  116203  000001                MOVB    1(R2),R3
 2618  021100  004767  003646                JSR     PC,DOUT         ;PRINT BAD CHAR
 2619  021104  012767  000240  157502        MOV     #240,TOB
 2620  021112  004767  003042                JSR     PC,TOG
 2621  021116  111203                MOVB    (R2),R3
 2622  021120  004767  003626                JSR     PC,DOUT         ;PRINT COMPLIMENT
 2623  021124  022122        PD5:    CMP     (R1)+,(R2)+     ;BUMP ADDRESS POINTERS
 2624  021126  005267  157642                INC     CRCNT           ;BUMP CHAR NUMBER
 2625  021132  005300                DEC     R0              ;SEE IF DONE
 2626  021134  001266                BNE     PD0             ;IF NOT: BR
 2627  021136  005767  157626                TST     PREFL           ;SEE IF PREAMBLE
```

```
 2628  021142  001402                        BEQ     PD6             ;IF SO: BR
 2629  021144  000167  177034                JMP     DCHKX           ;GO TO EXIT ROUTINE
 2630  021150  005267  157614        PD6:    INC     PPEFL           ;SET PREAMBLE FLAG
 2631  021154  005067  157440                CLR     HDRFL           ;CLEAR HEADER FLAG
 2632  021160  005067  157618                CLR     CRCNT           ;CLEAR CHAR CNTR
 2633  021164  005067  157576                CLR     DERFL           ;CLEAR DATA ERROR FLAG
 2634  021170  000167  000000                JMP     W1DCHK          ;GO CHECK WRAP 1 DATA
 2635
```

```
 2636
 2637                                        ;WAM 1 PE DATA CHECK*****
 2638
 2639  021174  012700  177400          W1DCHK: MOV     #-400,R0        ;SET NUMBER OF CHAR TO CHECK
 2640  021200  012701  035472                  MOV     #WBUFF,R1       ;SET WRITE DATA POINTER
 2641  021204  012702  036104                  MOV     #RBUFF,R2       ;SET READ DATA POINTER
 2642  021210  062702  000124                  ADD     #124,R2         ;POINT TO START OF DATA
 2643  021214  005767  157544                  TST     W2FLG           ;SEE IF WRAP 2
 2644  021220  001401                          BEQ     W1D0            ;IF NOT WAM 2: BR
 2645  021222  005302                          DEC     R2              ;RESET POINTER
 2646  021224  111105          W1D0:   MOVB    (R1),R5
 2647  021226  120512                  CMPB    R5,(R2)         ;CHECK DATA
 2648  021230  001007                  BNE     W1D1            ;IF NOT GOOD:^R
 2649  021232  005767  157526                  TST     W2FLG           ;SEE IF WRAP
 2650  021236  001001                          BNE     W1D0A           ;IF SO: BR
 2651  021240  105105                          COMB    R5              ;COMPLIMENT EXPT DATA
 2652  021242  120562  000001          W1D0A:  CMPB    R5,1(R2)        ;CHECK COMPLIMENT DATA
 2653  021246  001510                          BEQ     W1D3            ;IF GOOD: BR
 2654  021250  032777  020000  157312  W1D1:   BIT     #20000,@SWR     ;SEE IF PRINT INHIBIT
 2655  021256  001104                          BNE     W1D3            ;IF SO: BR
 2656  021260  005767  157334                  TST     HDRFL           ;SEE IF DONE HEADER
 2657  021264  001020                          BNE     W1D2            ;IF SO: BR
 2658  021266  016704  157330                  MOV     EMADDR,R4
 2659  021272  004767  002546                  JSR     PC,TTOUT        ;PRINT TEST HEADER
 2660  021276  012704  034740                  MOV     #WMSG16,R4
 2661  021302  004767  002536                  JSR     PC,TTOUT        ;PRINT BAD DATA TAG
 2662  021306  012704  035165                  MOV     #WMSG32,R4
 2663  021312  004767  002526                  JSR     PC,TTOUT        ;PRINT PATRN TAG
 2664  021316  016703  157460                  MOV     PATRN,R3
 2665  021322  004767  002660                  JSR     PC,OCTP         ;PRINT PATTERN NUMBER
 2666  021326  012767  000001  157264  W1D2:   MOV     #1,HDRFL        ;SET HEADER FLAG
 2667  021334  012704  034764                  MOV     #WMSG21,R4
 2668  021340  004767  002500                  JSR     PC,TTOUT        ;PRINT CHAR NUMBER TAG
 2669  021344  016703  157424                  MOV     CRCNT,R3
 2670  021350  004767  002632                  JSR     PC,OCTP         ;PRINT CHAR NUMBER
 2671  021354  012704  034752                  MOV     #WMSG17,R4
 2672  021360  004767  002460                  JSR     PC,TTOUT        ;PRNT GOOD TAG
 2673  021364  111105                          MOVB    (R1),R5
 2674  021366  110503                          MOVB    R5,R3           ;GET GOOD CHAR
 2675  021370  005767  157370                  TST     W2FLG           ;SEE IF WRAP 2
 2676  021374  001001                          BNE     W1D2A           ;IF SO: BR
 2677  021376  105103                          COMB    R3              ;ELSE COMPLIMENT CHAR
 2678  021400  004767  003346          W1D2A:  JSR     PC,DOUT         ;PRINT CHARACTER
 2679  021404  012767  000240  157282          MOV     #240,TOB
 2680  021412  004767  002542                  JSR     PC,TOG          ;SPACE
 2681  021416  110503                          MOVB    R5,R3
 2682  021420  004767  003326                  JSR     PC,DOUT         ;PRINT CHAR
 2683  021424  012704  034757                  MOV     #WMSG20,R4
 2684  021430  004767  002410                  JSR     PC,TTOUT        ;PRINT BAD TAG
 2685  021434  116203  000001                  MOVB    1(R2),R3
 2686  021440  004767  003306                  JSR     PC,DOUT         ;PRINT BAD CHAR
 2687  021444  012767  000240  157142          MOV     #240,TOB
 2688  021452  004767  002502                  JSR     PC,TOG          ;SPACE
 2689  021456  111203                          MOVB    (R2),R3
 2690  021460  004767  003266                  JSR     PC,DOUT         ;PRINT CHAR
 2691  021464  005267  157276                  INC     DFBF            ;SET DATA F
```

```
 2692  021470  122122                 W1D3:   CMPB    (R1)+,(R2)+     ;BUMP ADDRESS
 2693  021472  105722                         TSTB    (R2)+           ;BUMP ADDRESS
 2694  021474  005267  157274                 INC     CRCNT           ;BUMP CHAR CNTR
 2695  021500  000406                         BR      W1D5
 2696  021502  005767  157256         W1D4:   TST     W2FLG           ;SEE IF WRAP 2
 2697  021506  001401                         BEQ     W1D4A           ;IF NOT; BR
 2698  021510  000207                         RTS     PC              ;ELSE RETURN
 2699  021512  000167  177126         W1D4A:  JMP     PSCHK           ;GO CHECK POSTAMBLE
 2700  021516  005200                 W1D5:   INC     R0
 2701  021520  001770                         BEQ     W1D4
 2702  021522  000167  177476                 JMP     W1D0
 2703
 2704                                         ;PREAMBLE/POSTAMBLE GENERATE SUBROUTINE**********
 2705
 2706  021526  012700  000050         PPGEN:  MOV     #50,R0          ;SET SIZE OF PREAMBLE
 2707  021532  012701  035226                 MOV     #PRE,R1
 2708  021536  005721                         TST     (R1)+           ;SET ADDRESS OF PRE
 2709  021540  012721  177400         PPG0:   MOV     #177400,(R1)+   ;FILL TABLE
 2710  021544  005300                         DEC     R0              ;SEE IF DONE
 2711  021546  001374                         BNE     PPG0            ;IF NOT; BR
 2712  021550  012701  035350                 MOV     #POST,R1        ;SET ADDRESS OF POST
 2713  021554  012700  000050                 MOV     #50,R0          ;SET SIZE OF POST
 2714  021560  012721  000377                 MOV     #377,(R1)+      ;SET SYNC CHAR
 2715  021564  012721  177400         PPG1:   MOV     #177400,(R1)+   ;FILL TABLE
 2716  021570  005300                         DEC     R0              ;SEE IF DONE
 2717  021572  001374                         BNE     PPG1            ;IF NOT; BR
 2718  021574  000207                         RTS     PC              ;RETURN
```

```
2719
2720                                                    ;END OF RECORD FORCE SUBROUTINE*****
2721
2722  021576  005267  157074          EORPA:  INC     TEMP2           ;SET WRAP FLAG
2723  021602  017700  156726          EORP1:  MOV     @MR,R0          ;GET MAINT REG
2724  021606  042700  000036                  BIC     #36,R0          ;CLEAR CURRENT OP CODE
2725  021612  052700  000024                  BIS     #24,R0          ;SET EOR CLEAR OP CODE
2726  021616  010077  156712                  MOV     R0,@MR          ;DO EOR
2727  021622  042777  000037  156704          BIC     #37,@MR         ;CLEAR EOR AND MM
2728  021630  005000                          CLR     R0
2729  021632  012701  000002                  MOV     #2,R1
2730  021636  032777  000001  156644  EORP11: BIT     #1,@C1          ;SEE IF GO GONE
2731  021644  001431                          BEQ     EORP2           ;IF SO: BR
2732  021646  005300                          DEC     R0
2733  021650  001372                          BNE     EORP1           ;AWAIT GO RESET
2734  021652  005301                          DEC     R1
2735  021654  001370                          BNE     EORP1
2736  021656  032777  020000  156704          BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
2737  021664  001021                          BNE     EORP2           ;IF SO: BR
2738  021666  005767  156726                  TST     HDRFL           ;SEE IF DONE HEADER
2739  021672  001004                          BNE     EORP1A          ;IF SO: BR
2740  021674  016704  156722                  MOV     EMADDR,R4
2741  021700  004767  002140                  JSR     PC,TTOUT        ;PRINT HEADER
2742  021704  012704  035130          EORP1A: MOV     #MM&G31,R4
2743  021710  004767  002130                  JSR     PC,TTOUT        ;PRINT EOR GO BIT ERROR
2744  021714  032777  100000  156646          BIT     #100000,@SWR    ;SEE IF HALT ON ERROR
2745  021722  001402                          BEQ     EORP2           ;IF NOT: BR
2746  021724  004767  003232                  JSR     PC,@TOP
2747  021730  005767  156742          EORP2:  TST     TEMP2           ;SEE IF WAM
2748  021734  001014                          BNE     EORPX           ;IF NOT: BR
2749  021736  032777  000200  156624          BIT     #200,@SWR       ;SEE IF STATUS CHECK
2750  021744  001002                          BNE     EORP3           ;IF NOT: BR
2751  021746  004767  175154                  JSR     PC,@STCK        ;ELSE GO CHECK STATUS
2752  021752  032777  000400  156610  EORP3:  BIT     #400,@SWR       ;SEE IF DATA CHECK
2753  021760  001002                          BNE     EORPX           ;IF NOT: BR
2754  021762  004767  175430                  JSR     PC,DCHK         ;ELSE GO CHECK DATA
2755  021766  005067  156704          EORPX:  CLR     TEMP2           ;CLEAR FLAG
2756  021772  000207                          RTS     PC              ;RETURN
2757
```

```
 2758                                          ;LOGIC TEST ADDRESSING ERROR SUBROUTINE**********
 2759
 2760  021774  005067  156716          LTGER3: CLR     EXFL
 2761  022000  012767  027225  156664          MOV     #MSG51,ERADD
 2762  022006  012767  000001  156732  LTGER0: MOV     #1,ADDFL        ;SET NO ADDRESS FLAG
 2763  022014  005067  156646          LTGER:  CLR     PFLG            ;CLEAR PRINT FLAG
 2764  022020  032777  020000  156542          BIT     #20000,@SWR     ;SEE IF SHOULD PRINT
 2765  022026  001402                          BEQ     LTGA            ;IF SO: BR
 2766  022030  000167  000224                  JMP     LTGX            ;ELSE GO TO EXIT
 2767  022034  005767  156560          LTGA:   TST     HDRFL           ;SEE IF PRINTED HEADER
 2768  022040  001004                          BNE     LTGA1           ;IF SO: BR
 2769  022042  016704  156554                  MOV     EMADDR,R4
 2770  022046  004767  001772                  JSR     PC,TTOUT        ;PRINT TEST HEADER
 2771  022052  012767  000001  156540  LTGA1:  MOV     #1,HDRFL        ;SET HEADER FLAG
 2772  022060  016704  156606                  MOV     ERADD,R4
 2773  022064  004767  001754                  JSR     PC,TTOUT        ;PRINT CONDITION ERROR
 2774  022070  005767  156652                  TST     ADDFL
 2775  022074  001003                          BNE     LTGA2
 2776  022076  010103                          MOV     R1,R3
 2777  022100  004767  002102                  JSR     PC,OCTP         ;PRINT ADDRESS
 2778  022104  005767  156606          LTGA2:  TST     EXFL
 2779  022110  001412                          BEQ     LTGC            ;IF NO STATUS: BR
 2780  022112  012704  025524                  MOV     #MSG6,R4
 2781  022116  022767  000001  156572          CMP     #1,EXFL         ;EXPT-NOT RCVD
 2782  022124  001402                          BEQ     LTGB
 2783  022126  012704  025543                  MOV     #MSG7,R4        ;RCVD-NOT EXPT
 2784  022132  004767  001706          LTGB:   JSR     PC,TTOUT        ;PRINT STATUS
 2785  022136  005267  156524          LTGC:   INC     PFLG
 2786  022142  005767  156600                  TST     ADDFL           ;SEE IF ADD TST
 2787  022146  001438                          BEQ     LTGD            ;IF SO: BR
 2788  022150  005767  156570                  TST     T24FL           ;SEE IF TEST 24
 2789  022154  001423                          BEQ     LTGC0           ;IF NOT: BR
 2790  022156  012704  035060                  MOV     #WMSG27,R4
 2791  022162  004767  001656                  JSR     PC,TTOUT        ;PRINT DATA TAG
 2792  022166  012704  026022                  MOV     #MSG12,R4
 2793  022172  004767  001646                  JSR     PC,TTOUT        ;PRINT EXPT TAG
 2794  022176  012703  177777                  MOV     #-1,R3
 2795  022202  004767  001770                  JSR     PC,OCTPE        ;PRINT EXPT
 2796  022206  012704  026031                  MOV     #MSG13,R4
 2797  022212  004767  001626                  JSR     PC,TTOUT        ;PRINT RCVD TAG
 2798  022216  010103                          MOV     R1,R3           ;GET RCVD
 2799  022220  004767  001752                  JSR     PC,OCTPE        ;PRINT RCVD
 2800  022224  004767  000102          LTGC0:  JSR     PC,REGP         ;PRINT REGISTERS
 2801  022230  032777  010000  156332  LTGD:   BIT     #10000,@SWR
 2802  022236  001010                          BNE     LTGX
 2803  022240  012704  026073                  MOV     #MSG16,R4
 2804  022244  004767  001574                  JSR     PC,TTOUT
 2805  022250  016703  156426                  MOV     ITCNT,R3        ;PRINT ITERATION
 2806  022254  004767  001726                  JSR     PC,OCTP
 2807  022260  005777  156304          LTGX:   TST     @SWR
 2808  022264  100002                          BPL     LTGXA           ;IF NOT STOP ON ERROR: BR
 2809  022266  004767  002670                  JSR     PC,STOP
 2810  022272  005767  156370          LTGXA:  TST     PFLG
 2811  022276  001006                          BNE     LTGXX           ;IF PRINTED: BR
 2812  022300  032777  020000  156262          BIT     #20000,@SWR
 2813  022306  001002                          BNE     LTGXX           ;IF ERROR: BR
```

```
2814  022310  000167  177520          JMP     LTGA            ;ELSE GO PRINT ERROR
2815  022314  005067  156426  LTGXX:  CLR     ADDFL           ;CLEAR ADDRESS FLAG
2816  022320  005067  156372          CLR     EXFL
2817  022324  000167  001044          JMP     SCOPE
2818  022330  000207                  RTS     PC              ;EXIT
2819
2820                                  ;SUBROUTINE TO PRINT MAJOR REGISTERS**********
2821
2822  022332  012704  027050  REGP:   MOV     #MSG46,R4
2823  022336  004767  001502          JSR     PC,TTOUT        ;PRINT REGISTER HEADER
2824  022342  017703  156142          MOV     @C1,R3
2825  022346  004767  001624          JSR     PC,OCTPE
2826  022352  017703  156134          MOV     @WC,R3
2827  022356  004767  001614          JSR     PC,OCTPE
2828  022362  017703  156126          MOV     @BA,R3
2829  022366  004767  001604          JSR     PC,OCTPE
2830  022372  017703  156120          MOV     @FC,R3
2831  022376  004767  001574          JSR     PC,OCTPE
2832  022402  017703  156112          MOV     @CS,R3
2833  022406  004767  001564          JSR     PC,OCTPE
2834  022412  017703  156104          MOV     @DS,R3
2835  022416  004767  001554          JSR     PC,OCTPE                ;PRINT REGISTERS
2836  022422  017703  156076          MOV     @ER,R3
2837  022426  004767  001544          JSR     PC,OCTPE
2838  022432  017703  156070          MOV     @AS,R3
2839  022436  004767  001534          JSR     PC,OCTPE
2840  022442  017703  156066          MOV     @MR,R3
2841  022446  004767  001524          JSR     PC,OCTPE
2842  022452  017703  156064          MOV     @TC,R3
2843  022456  004767  001514          JSR     PC,OCTPE
2844  022462  000207                  RTS     PC
2845
2846
```

```
 2847                                          ;DRIVE CLEAR SUBROUTINE*********
 2848
 2849  022464  012704  040000          DRVCLR: MOV     #40000,R4
 2850  022470  005304                  DCD:    DEC     R4
 2851  022472  001376                          BNE     DCD             ;DELAY
 2852  022474  005067  156166                  CLR     PFLG
 2853  022500  004767  000222                  JSR     PC,ATTN         ;GO SEE OF ATTN SET
 2854  022504  012777  000011  155776          MOV     #11,@C1         ;ISSUE DRIVE CLEAR
 2855  022512  005000                          CLR     R0
 2856  022514  032777  000200  156000  DCA:    BIT     #200,@DS        ;SEE IF DRY
 2857  022522  001002                          BNE     DCA0
 2858  022524  005300                          DEC     R0
 2859  022526  001372                          BNE     DCA             ;WAIT FOR DRY
 2860  022530  032777  040000  155764  DCA0:   BIT     #40000,@DS      ;SEE IF ERR RESET
 2861  022536  001024                          BNE     DCE             ;IF NOT: BR
 2862  022540  005777  155760                  TST     @ER             ;SEE IF ERROR REGISTER RESET
 2863  022544  001021                          BNE     DCE             ;IF NOT: BR
 2864  022546  005777  155750                  TST     @DS             ;SEE IF ATA RESET
 2865  022552  100416                          BMI     DCE             ;IF NOT: BR
 2866  022554  012703  000001                  MOV     #1,R3           ;SET TEST BIT
 2867  022560  016704  156040                  MOV     DRVN,R4         ;GET DRIVE NUMBER
 2868  022564  005704                          TST     R4              ;SEE IF DRIVE 0
 2869  022566  001404                          BEQ     DCC             ;IF SO: BR
 2870  022570  000241                  DCB:    CLC
 2871  022572  006103                          ROL     R3              ;POSITION TEST BIT PER DRIVE NUMBER
 2872  022574  005304                          DEC     R4              ;SEE IF DONE
 2873  022576  001374                          BNE     DCB             ;IF NOT: BR
 2874  022600  030377  155722          DCC:    BIT     R3,@AS          ;SEE IF ATTEN IS RESET
 2875  022604  001001                          BNE     DCE             ;IF NOT: BR
 2876  022606  000207                          RTS     PC              ;RETURN
 2877  022610  032777  020000  155752  DCE:    BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
 2878  022616  001017                          BNE     DCEX            ;IF SO: BR
 2879  022620  005767  155774                  TST     HDRFL           ;SEE IF PRINT HEADER
 2880  022624  001004                          BNE     DCEA            ;IF NOT: BR
 2881  022626  016704  155770                  MOV     EMADDR,R4
 2882  022632  004767  001206                  JSR     PC,TTOUT        ;PRINT HEADER
 2883  022636  012704  027154          DCEA:   MOV     #MSG47,R4
 2884  022642  004767  001176                  JSR     PC,TTOUT        ;PRINT DRIVE CLEAR ERROR
 2885  022646  004767  177460                  JSR     PC,REGP         ;PRINT REGISTERS
 2886  022652  005267  156010                  INC     PFLG            ;SET PRINTED FLAG
 2887  022656  005777  155706          DCEX:   TST     @SWR            ;SEE IF HALT ON ERROR
 2888  022662  100002                          BPL     DCEXA           ;IF NOT: BR
 2889  022664  004767  002272                  JSR     PC,STOP
 2890  022670  005767  155772          DCEXA:  TST     PFLG            ;SEE IF HAVE PRINTED
 2891  022674  001006                          BNE     DCEXX           ;IF SO: BR
 2892  022676  032777  020000  155664          BIT     #20000,@SWR     ;SEE IF SHOULD PRINT
 2893  022704  001002                          BNE     DCEXX           ;IF NOT: BR
 2894  022706  000167  177676                  JMP     DCE             ;ELSE PRINT THIS ERROR
 2895  022712  012767  022464  155772  DCEXX:  MOV     #DRVCLR,SCOLP   ;SET SCOPE LOOP ADDRESS
 2896  022720  000167  000450                  JMP     SCOPE           ;GO DO SCOPE LOOP
 2897  022724  000207                          RTS     PC              ;RETURN
```

```
 2898                                          ;COMPOSITE ERROR CHECK SUBROUTINE**********
 2899
 2900  022726  005777  155570        ATTN:   TST     @DS             ;SEE IF ATA SET
 2901  022732  001005                        BNE     ATTA            ;IF SO: BR
 2902  022734  012767  026512  155736        MOV     #MSG32,TEMP3
 2903  022742  000167  000064                JMP     ATTP            ;ELSE PRINT ERROR
 2904  022746  032777  040000  155546  ATTA: BIT     #40000,@DS      ;SEE IF COMPOSITE ERROR SET
 2905  022754  001005                        BNE     ATTB            ;IF SO: BR
 2906  022756  012767  026474  155714        MOV     #MSG31,TEMP3
 2907  022764  000167  000042                JMP     ATTP            ;ELSE PRINT ERROR
 2908  022770  012703  000001        ATTB: MOV     #1,R3           ;SET TEST BIT
 2909  022774  012767  026530  155676        MOV     #MSG33,TEMP3
 2910  023002  016704  155616                MOV     DRVN,R4         ;GET DRIVE NUMBER
 2911  023006  005704                        TST     R4              ;SEE IF DRIVE 0
 2912  023010  001404                        BEQ     ATTD            ;IF SO: BR
 2913  023012  000241        ATTC: CLC
 2914  023014  006103                        ROL     R3              ;POSITION TEST BIT
 2915  023016  005304                        DEC     R4              ;SEE IF DONE
 2916  023020  001374                        BNE     ATTC            ;IF NOT: BR
 2917  023022  030377  155500        ATTD: BIT     R3,@AS          ;SEE IF ATTEN SUMMARY SET
 2918  023026  001401                        BEQ     ATTP            ;IF NOT: BR
 2919  023030  000207                        RTS     PC              ;ELSE RETURN
 2920  023032  032777  020000  155530  ATTP: BIT     #20000,@SWR     ;SEE IF PRINT INHIBIT
 2921  023040  001021                        BNE     ATTX            ;IF SO: BR
 2922  023042  005767  155552                TST     HDRFL           ;SEE IF DONE HEADER
 2923  023046  001004                        BNE     ATTPA           ;IF SO: BR
 2924  023050  016704  155546                MOV     EMADDR,R4
 2925  023054  004767  000764                JSR     PC,TTOUT        ;PRINT HEADER
 2926  023060  016704  155614        ATTPA: MOV     TEMP3,R4
 2927  023064  004767  000754                JSR     PC,TTOUT        ;PRINT ERROR TYPE
 2928  023070  004767  177236                JSR     PC,REGP         ;PRINT REGISTERS
 2929  023074  005267  155566                INC     PFLG            ;SET PRINT FLAG
 2930  023100  005267  155514                INC     HDRFL           ;SET HEADER FLAG
 2931  023104  005777  155460        ATTX: TST     @SWR            ;SEE IF HALT ON ERROR
 2932  023110  100002                        BPL     ATTXA           ;IF NOT: BR
 2933  023112  004767  002044                JSR     PC,STOP
 2934  023116  005767  155544        ATTXA: TST     PFLG            ;SEE IF DONE PRINT
 2935  023122  001006                        BNE     ATTXX           ;IF SO: BR
 2936  023124  032777  020000  155436        BIT     #20000,@SWR     ;SEE IF SHOULD PRINT
 2937  023132  001002                        BNE     ATTXX           ;IF NOT: BR
 2938  023134  000167  177672                JMP     ATTP            ;ELSE PRINT ERROR
 2939  023140  005067  155522        ATTXX: CLR     PFLG            ;CLEAR PRINT FLAG
 2940  023144  000207                        RTS     PC              ;RETURN
```

```
2941                                                ;LOGIC TEST REGISTER BIT ERROR SUBROUTIINE**********
2942
2943  023146  012767  000001  155562  LTGER2: MOV     #1,PEXFL        ;SET FLAG
2944  023154  005067  155586          LTGER1: CLR     PFLG            ;CLEAR PRINT FLAG
2945  023160  032777  020000  155482          BIT     #20000,@SWR     ;SEE IF PRINT ERRORS
2946  023166  001402                          BEQ     LTG1A           ;IF SO: BR
2947  023170  000167  000132                  JMP     LTG1X           ;ELSE GO TO EXIT
2948  023174  005767  155420          LTG1A:  TST     HDRFL           ;SEE IF PRINT HEADER
2949  023200  001004                          BNE     LTG1B           ;IF NOT: BR
2950  023202  016704  155414                  MOV     EMADDR,R4
2951  023206  004767  000632                  JSR     PC,TTOUT        ;PRINT HEADER
2952  023212  012767  000001  155400  LTG1B:  MOV     #1,HDRFL        ;SET FLAG
2953  023220  016704  155446                  MOV     ERADD,R4
2954  023224  004767  000614                  JSR     PC,TTOUT        ;PRINT ERROR CODE
2955  023230  005767  155502                  TST     PEXFL           ;SEE IF PRINT EXPT-RCVD
2956  023234  001016                          BNE     LTG1T           ;IF NOT: BR
2957  023236  012704  026022                  MOV     #MSG12,R4
2958  023242  004767  000576                  JSR     PC,TTOUT        ;PRINT EXPT TAG
2959  023246  010103                          MOV     R1,R3
2960  023250  004767  000732                  JSR     PC,OCTP         ;PRINT EXPT
2961  023254  012704  026031                  MOV     #MSG13,R4
2962  023260  004767  000560                  JSR     PC,TTOUT        ;PRINT RCVD TAG
2963  023264  010203                          MOV     R2,R3
2964  023266  004767  000714                  JSR     PC,OCTP         ;PRINT RCVD
2965  023272  032777  010000  155270  LTG1T:  BIT     #10000,@SWR
2966  023300  001010                          BNE     LTG1C
2967  023302  012704  026073                  MOV     #MSG16,R4
2968  023306  004767  000532                  JSR     PC,TTOUT
2969  023312  016703  155364                  MOV     ITCNT,R3
2970  023316  004767  000664                  JSR     PC,OCTP         ;PRINT ITERATION
2971  023322  005267  155340          LTG1C:  INC     PFLG
2972  023326  005777  155236          LTG1X:  TST     @SWR
2973  023332  100002                          BPL     LTG1X1          ;IF NOT STOP ON ERROR: BR
2974  023334  004767  001622                  JSR     PC,STOP
2975  023340  005767  155322          LTG1X1: TST     PFLG
2976  023344  001006                          BNE     LTG1XX          ;IF HAVE PRINTED: BR
2977  023346  032777  020000  155214          BIT     #20000,@SWR
2978  023354  001002                          BNE     LTG1XX          ;IF STILL NO PRINT: BR
2979  023356  000167  177612                  JMP     LTG1A           ;ELSE PRINT ERROR
2980  023362  005067  155350          LTG1XX: CLR     PEXFL           ;CLEAR EXPT-RCVD FLAG
2981  023366  000167  000002                  JMP     SCOPE           ;GO TO SCOPE
2982  023372  000207                          RTS     PC              ;RETURN
2983
```

```
2984
2985                                        ;SCOPE LOOP ON ERROR SUBROUTINE**********
2986
2987  023374  004767  001042        SCOPE:  JSR     PC,CKSWR        ;CHECK FOR CONTROL G
2988  023400  032777  040000  155162        BIT     #40000,@SWR     ;SEE IF LOOP ON ERROR
2989  023406  001001                        BNE     SCOPE1          ;IF SO: BR
2990  023410  000207                        RTS     PC              ;ELSE EXIT
2991  023412  005726                SCOPE1: TST     (SP)+           ;RESET STACK
2992  023414  000177  155272                JMP     @SCOLP          ;LOOP ON ERROR
2993
2994                                        ;TEST ITERATION SUBROUTINE**********
2995
2996  023420  032777  010000  155142  ITER:  BIT     #10000,@SWR     ;SEE IF ITERATIONS
2997  023426  001403                        BEQ     ITER1           ;IF SO: BR
2998  023430  005067  155246        ITER0:  CLR     ITCNT           ;CLEAR ITERATION COUNTER
2999  023434  000207                        RTS     PC              ;ELSE EXIT
3000  023436  005267  155240        ITER1:  INC     ITCNT           ;BUMP COUNTER
3001  023442  026767  155234  155136        CMP     ITCNT,ITAMT     ;SEE IF DONE ALL
3002  023450  001767                        BEQ     ITER0           ;IF SO: BR
3003  023452  005726                        TST     (SP)+           ;RESET STACK
3004  023454  017700  155234                MOV     @ITRLP,R0       ;SET ITERATION POINTER
3005  023460  000110                        JMP     (R0)            ;GO ITERATE
3006
3007                                        ;MANUAL INTERVENTION INHIBIT**********
3008
3009  023462  012704  026700        INMT:   MOV     #MSG43,R4
3010  023466  004767  000352                JSR     PC,TTOUT        ;GO PRINT INHIB MSG
3011  023472  004767  001464                JSR     PC,STOP
3012  023476  000167  156414                JMP     TSCD2           ;RETURN TO SCHED
3013
```

```
                                        ;INITIALIZE SUBROUTINE**********
                                        
023502  012777  000040  155010  INIT1:  MOV     #40,@CS         ;INIT
023510  016777  155110  155002  INIT2:  MOV     DRVN,@CS        ;SELECT DRIVE
023516  016777  155142  155016          MOV     SLVN,@TC        ;SELECT SLAVE
023524  000207                          RTS     PC              ;RETURN
                                        
                                        ;MANUAL INSTRUCTION SUBROUTINE**********
                                        
023526  004767  000312          INST:   JSR     PC,TTOUT        ;PRINT INSTRUCTION
023532  012704  033212                  MOV     #MMSG0,R4
023536  004767  000302                  JSR     PC,TTOUT        ;PRINT REPLY
023542  012705  000700                  MOV     #TEMP3,R5
023546  012701  000001                  MOV     #1,R1
023552  012702  177777                  MOV     #-1,R2
023556  012703  000000                  MOV     #0,R3
023562  004767  000020                  JSR     PC,TIR          ;AWAIT REPLY
023566  000207                          RTS     PC              ;EXIT
                                        
                                        ;MAG TAPE INTERRUPT HANDLER**********
                                        
023570  022626                  MTINT:  CMP     (SP)+,(SP)+     ;RESET STACK POINTER
023572  000240                          NOP
023574  000240                          NOP
023576  000177  155066                  JMP     @RTRN           ;RETURN TO CALLER
                                        
                                        ;TTY INTERRUPT HANDLER**********
                                        
023602                          TTINT:
023602  000240                          NOP
023604  000002                          RTI
```

```
                                        ;*************************************************
                                        ;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
                                        ;*************************************************
023606  005067  155062          TTR:    CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
023612  005000                          CLR     R0
023614  004767  000152          TTR0:   JSR     PC,TTIN         ;GO READ CHARACTER
023620  122767  000215  154770          CMPB    #215,TIB        ;SEE IF CR
023626  001005                          BNE     TTR1            ;IF NOT: BR
023630  005767  155040                  TST     TEMP1           ;SEE IF FIRST CHARACTER
023634  001446                          BEQ     TTR5            ;IF SO: BR
023636  000167  000066                  JMP     TTR2            ;ELSE GO LOAD VALUE
023642  122767  000260  154746  TTR1:   CMPB    #260,TIB        ;SEE IF CHAR IS LESS THAN 0
023650  101402                          BLOS    TTR1A           ;IF NOT: BR
023652  000167  000076                  JMP     TINER           ;ELSE GO TO ERROR
023656  122767  000270  154732  TTR1A:  CMPB    #270,TIB        ;SEE IF CHAR IS GREATER THAN 7
023664  101002                          BHI     TTR1B           ;IF NOT: BR
023666  000167  000062                  JMP     TINER           ;ELSE GO TO ERROR
023672  005267  154776          TTR1B:  INC     TEMP1           ;SET FIRST CHARACTER FLAG
023676  000241                          CLC
023700  006100                          ROL     R0
023702  000241                          CLC
023704  006100                          ROL     R0              ;SHIFT 3 LEFT
023706  000241                          CLC
023710  006100                          ROL     R0
023712  042767  177770  154676          BIC     #177770,TIB     ;STRIP ASCII
023720  056700  154672                  BIS     TIB,R0          ;LOAD CHARACTER
023724  005301                          DEC     R1              ;SEE IF DONE
023726  001332                          BNE     TTR0            ;IF NOT: BR
023730  020002                  TTR2:   CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
023732  101402                          BLOS    TTR3            ;IF NOT: BR
023734  000167  000014                  JMP     TINER           ;ELSE GO TO ERROR
023740  020300                  TTR3:   CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
023742  101402                          BLOS    TTR4            ;IF NOT: BR
023744  000167  000004                  JMP     TINER           ;ELSE GO TO ERROR
023750  010015                  TTR4:   MOV     R0,(R5)         ;LOAD VALUE
023752  000207                  TTR5:   RTS     PC              ;EXIT
```

```
3098
3099                                    ;TTY ENTRY ERROR SUBROUTINE**********
3100
3101  023754  012704  026640    TINER:  MOV     #MSG40,R4
3102  023760  004767  000060            JSF     PC,TTOUT        ;PRINT?
3103  023764  162716  000020            SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
3104  023770  000207                    RTS     PC              ;REDO VALUE ENTRY
3105
3106                                    ;TTY READ SUBROUTINE**********
3107
3108  023772  005077  154574    TTIN:   CLR     @TKS
3109  023776  005077  154572            CLR     @TKB
3110  024002  005067  154618            CLR     TIB
3111  024006  005277  154560            INC     @TKS
3112  024012  105777  154554    TTIN1:  TSTB    @TKS
3113  024016  100375                    BPL     TTIN1
3114  024020  017767  154550  154570    MOV     @TKB,TIB
3115  024026  105777  154544    TTIN2:  TSTB    @TPS
3116  024032  100375                    BPL     TTIN2
3117  024034  116777  154556  154536    MOVB    TIB,@TPB
3118  024042  000207                    RTS     PC
3119
3120                                    ;TTY OUTPUT SUBROUTINE**********
3121
3122  024044  112467  154544    TTOUT:  MOVB    (R4)+,TOB
3123  024050  122767  000043  154536    CMPB    #43,TOB
3124  024056  001446                    BEQ     TEX
3125  024060  122767  000045  154526    CMPB    #45,TOB
3126  024066  001403                    BEQ     TCRLF
3127  024070  004767  000064            JSR     PC,TOG
3128  024074  000763                    BR      TTOUT
3129  024076  112767  000015  154510  TCRLF:  MOVB    #15,TOB
3130  024104  004767  000050            JSR     PC,TOG
3131  024110  012703  000004            MOV     #4,R3
3132  024114  005067  154474    TCRLFA: CLR     TOB
3133  024120  004767  000034            JSR     PC,TOG
3134  024124  005303                    DEC     R3
3135  024126  001372                    BNE     TCRLFA          ;DO FILLERS
3136  024130  112767  000012  154456    MOVB    #12,TOB
3137  024136  004767  000016            JSR     PC,TOG
3138  024142  105767  000272            TSTB    RDSW
3139  024146  100401                    BMI     1$
3140  024150  000735                    BR      TTOUT
3141  024152  005067  000262    1$:     CLR     RDSW
3142  024156  000406                    BR      TEX
3143  024160  105777  154412    TOG:    TSTB    @TPS
3144  024164  100375                    BPL     TOG
3145  024166  116777  154422  154404    MOVB    TOB,@TPB
3146  024174  000207            TEX:    RTS     PC
3147
3148
3149                                    ;OCTAL OUTPUT SUBROUTINE**********
3150
3151  024176  012767  000001  000226  OCTPE:  MOV     #1,OFL
3152  024204  000402                    BR      OCTPE1
3153  024206  005067  000220    OCTP:   CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
```

```
 3154  024212  010304                OCTPE1: MOV     R3,R4           ;SEE IF NUMBER IS ZERO
 3155  024214  001007                        BNE     OCTP0           ;IF NOT ZERO: BR
 3156  024216  005767  000210                TST     OFL             ;SEE IF PRINT ALL 0
 3157  024222  001004                        BNE     OCTP0           ;IF SO: BR
 3158  024224  004767  000162                JSR     PC,OCTPG1       ;ELSE PRINT ZERO
 3159  024230  000167  000120                JMP     OCTP3           ;SPACE AND EXIT
 3160  024234  032704  100000        OCTP0:  BIT     #100000,R4      ;SEE IF MSD = 1
 3161  024240  001406                        BEQ     OCTP1           ;IF NOT: BR
 3162  024242  012704  000001                MOV     #1,R4
 3163  024246  004767  000116                JSR     PC,OCTPG        ;PRINT 1
 3164  024252  000167  000006                JMP     OCTP2
 3165  024256  005004                OCTP1:  CLR     R4
 3166  024260  004767  000104                JSR     PC,OCTPG        ;PRINT 0
 3167  024264  010304                OCTP2:  MOV     R3,R4
 3168  024266  006004                        ROR     R4
 3169  024270  006004                        ROR     R4
 3170  024272  006004                        ROR     R4              ;POSITION DIGIT
 3171  024274  006004                        ROR     R4
 3172  024276  000304                        SWAB    R4
 3173  024300  004767  000064                JSR     PC,OCTPG        ;PRINT DIGIT 2
 3174  024304  010304                        MOV     R3,R4
 3175  024306  006004                        ROR     R4
 3176  024310  000304                        SWAB    R4
 3177  024312  004767  000052                JSR     PC,OCTPG        ;PRINT DIGIT 3
 3178  024316  010304                        MOV     R3,R4
 3179  024320  006104                        ROL     R4
 3180  024322  006104                        ROL     R4
 3181  024324  000304                        SWAB    R4
 3182  024326  004767  000036                JSR     PC,OCTPG        ;PRINT DIGIT 4
 3183  024332  010304                        MOV     R3,R4
 3184  024334  006004                        ROR     R4
 3185  024336  006004                        ROR     R4
 3186  024340  006004                        ROR     R4
 3187  024342  004767  000022                JSR     PC,OCTPG
 318'  024346  010304                        MOV     R3,R4
 3189  024350  004767  000014                JSR     PC,OCTPG        ;PRINT DIGIT 5
 3190  024354  012767  000240  154232  OCTP3: MOV     #240,TOB
 3191  024362  004767  177572                JSR     PC,TOG          ;PRINT SPACE
 3192  024366  000207                        RTS     PC              ;EXIT
 3193  024370  042704  177770        OCTPG:  BIC     #177770,R4
 3194  024374  001004                        BNE     OCTPG0
 3195  024376  005767  000030                TST     OFL
 3196  024402  001001                        BNE     OCTPG0
 3197  024404  000207                        RTS     PC
 3198  024406  005267  000020        OCTPG0: INC     OFL
 3199  024412  052704  000260        OCTPG1: BIS     #260,R4
 3200  024416  010467  154172                MOV     R4,TOB
 3201  024422  004767  177532                JSR     PC,TOG
 3202  024426  010304                        MOV     R3,R4
 3203  024430  000207                        RTS     PC
 3204  024432  000000                OFL:    0                       ;FIRST CHAR FLAG
 3205
 3206
 3207                                        ;CHECK SWITCH REGISTER ROUTINE, CHECKS FOR ^G TO ALLOW CHANGING
 3208                                        ;OF LOC.176.
 3209                                        ;CALL IS BY JSR PC,SWR
```

```
3210                                        ;LOCATIONS USED:
3211  024434  000000                        TEMPST: ,WORD   0
3212  024436  000000                        COUNT:  ,WORD   0
3213  024440  000000                        RDSW:   ,WORD   0
3214  024442  022767  000176  154120 CKSWR: CMP     #SWREG,SWR              ;SOFTWARE SWITCH REG PRESENT
3215  024450  001037                        BNE     OUT                     ;NO, GET OUT
3216  024452  105777  154114                TSTB    @TKS                    ;YES,WAIT FOR
3217  024456  100034                        BPL     OUT                     ;READY, GET CHARACTER
3218  024460  017767  154110  154130        MOV     @TKB,TIB                ;AND STRIP OFF
3219  024466  042767  177600  154122        BIC     #177600,TIB             ;THE GARBAGE
3220  024474  022767  000007  154114        CMP     #7,TIB          ;IS IT A <^G>
3221  024502  001022                        BNE     OUT
3222  024504  012704  035175                MOV     #$CNTG,R4
3223  024510  004767  177330                JSR     PC,TTOUT
3224  024514  012704  035201        CNTLU:  MOV     #$MSWR,R4
3225  024520  004767  177320                JSR     PC,TTOUT
3226  024524  017703  154040                MOV     @SWR,R3
3227  024530  004767  177442                JSR     PC,OCTPE
3228  024534  012704  035210                MOV     #$MNEW,R4
3229  024540  004767  177300                JSR     PC,TTOUT
3230  024544  004767  000002                JSR     PC,$READ                ;GO READ A LINE
3231  024550  000207                OUT:    RTS     PC                      ;RETURN TO MAIN BODY OF PROGRAM
3232
3233  024552  005067  177656        $READ:  CLR     TEMPST
3234  024556  012767  000007  177652        MOV     #7,COUNT
3235  024564  004767  177202          1$:   JSR     PC,TTIN         ;GO  READ A CHARACTER
3236  024570  042767  177600  154020        BIC     #177600,TIB     ;STRIP OFF GARBAGE
3237  024576  122767  000025  154012        CMPB    #25,TIB         ;IS IT A ^U?
3238  024604  001002                        BNE     2$              ;BRANCH IF NOT
3239  024606  005726                  3$:   TST     (SP)+           ;POP THE STACK
3240  024610  000741                        BR CNTLU                ;START OVER
3241  024612  122767  000015  153776  2$:   CMPB    #15,TIB         ;IS IT A <CR>?
3242  024620  001013                        BNE     4$              ;BRANCH IF NOT
3243  024622  012767  000200  177610        MOV     #200,RDSW
3244  024630  004767  177242                JSR     PC,TCRLF        ;ECHO IT WITH <LF>
3245  024634  022767  000007  177574        CMP     #7,COUNT        ;WAS IT FIRST CHARACTER
3246  024642  001037                        BNE     7$              ;CHANGE SWR IF NOT FIRST ONE
3247  024644  005726                  8$:   TST     (SP)+           ;POP THE STACK
3248  024646  000740                        BR      OUT             ;GET OUT
3249  024650  122767  000060  153740  4$:   CMPB    #60,TIB
3250  024656  003004                        BGT     5$
3251  024660  122767  000067  153730        CMPB    #67,TIB
3252  024666  002005                        BGE     6$
3253  024670  012704  035220          5$:   MOV     #$QUEST,R4
3254  024674  004767  177144                JSR     PC,TTOUT
3255  024700  000742                        BR      3$              ;START OVER IF NOT LEGAL CHARACTER
3256  024702  006367  177526          6$:   ASL     TEMPST
3257  024706  006367  177522                ASL     TEMPST
3258  024712  006367  177516                ASL     TEMPST
3259  024716  142767  000060  153672        BICB    #60,TIB         ;GET NITTY-GRITTY
3260  024724  156767  153666  177502        BISB    TIB,TEMPST
3261  024732  005367  177500                DEC     COUNT           ;ONLY WANT 6   DIGITS
3262  024736  001754                        BEQ     5$
3263  024740  000711                        BR      1$
3264  024742  016777  177466  153620  7$:   MOV     TEMPST,@SWR     ;CHANGE SWITCH REGISTER CONTENTS
3265  024750  000735                        BR      8$
```

```
3266
3267                                            ;DATA CHARACTER OUTPUT SUBROUTINE**********
3268
3269  024752  005067  153636          DOUT:   CLR     TOB
3270  024756  012704  000010                  MOV     #10,R4          ;SET NUMBER TO PRINT
3271  024762  110367  153626                  MOVB    R3,TOB
3272  024766  105777  153604          DOUT1:  TSTB    @TPS
3273  024772  100375                          BPL     DOUT1
3274  024774  132767  000200  153612          BITB    #200,TOB
3275  025002  001404                          BEQ     DOUT2
3276  025004  012777  000061  153566          MOV     #061,@TPB
3277  025012  000403                          BR      DOUT3
3278  025014  012777  000060  153556  DOUT2:  MOV     #060,@TPB
3279  025022  006167  153566          DOUT3:  ROL     TOB
3280  025026  005304                          DEC     R4
3281  025030  001356                          BNE     DOUT1
3282  025032  000207                          RTS     PC
3283  025034  016703  153640          DOUTD:  MOV     TEMP3,R3
3284  025040  000303                          SWAB    R3
3285  025042  004767  177784                  JSR     PC,DOUT
3286  025046  016703  153626                  MOV     TEMP3,R3
3287  025052  004767  177674                  JSR     PC,DOUT
3288  025056  000207                          RTS     PC
3289
3290                                            ;TU16 SERIAL NUMBER PRINT SUBROUTINE**********
3291
3292  025060  010304                  SNPT:   MOV     R3,R4
3293  025062  000304                          SWAB    R4
3294  025064  006004                          ROR     R4
3295  025066  006004                          ROR     R4
3296  025070  006004                          ROR     R4
3297  025072  006004                          ROR     R4              ;GET FIRST DIGIT
3298  025074  004767  000036                  JSR     PC,SNPG         ;PRINT
3299  025100  010304                          MOV     R3,R4
3300  025102  000304                          SWAB    R4              ;GET SECOND DIGIT
3301  025104  004767  000026                  JSR     PC,SNPG         ;PRINT
3302  025110  010304                          MOV     R3,R4
3303  025112  006004                          ROR     R4
3304  025114  006004                          ROR     R4
3305  025116  006004                          ROR     R4
3306  025120  006004                          ROR     R4
3307  025122  004767  000010                  JSR     PC,SNPG         ;PRINT THIRD DIGIT
3308  025126  010304                          MOV     R3,R4
3309  025130  004767  000002                  JSR     PC,SNPG         ;PRINT FOURTH DIGIT
3310  025134  000207                          RTS     PC              ;EXIT
3311  025136  012767  000260  153450  SNPG:   MOV     #260,TOB        ;SET BASE = 0
3312  025144  042704  177760                  BIC     #177760,R4      ;MASK DIGIT
3313  025150  050467  153440                  BIS     R4,TOB          ;SET ASCII
3314  025154  004767  177000                  JSR     PC,TOG          ;TYPE DIGIT
3315  025160  000207                          RTS     PC              ;RETURN
3316
3317                                            ;HALT HANDLER**********
3318
3319  025162  000000                  STOP:   HALT
3320  025164  004767  177252                  JSR     PC,CKSWR        ;CHECK FOR CONTROL G
3321  025170  000207                          RTS     PC
```

```
 3322                                                ;MESSAGE TABLE**********
 3323
 3324  025172  022445  046524  031060  MSG1:   .ASCII  /%%TM02-TU16 CONTROL LOGIC TEST (DZTUC-E)%/
 3325  025200  052055  030525  020066
 3326  025206  047503  052116  047522
 3327  025214  020114  047514  044507
 3328  025222  020103  042524  052123
 3329  025230  024040  055104  052524
 3330  025236  026503  024505     043
 3331  025243     105  052116  051105          .ASCII  /ENTER CONDITIONS IN OCTAL%%/
 3332  025250  041440  047117  044504
 3333  025256  044524  047117  020123
 3334  025264  047111  047440  052103
 3335  025272  046101  021445
 3336  025276  042045  044522  042526  MSG2:   .ASCII  /%DRIVE NUMBER %/
 3337  025304  047040  046525  042502
 3338  025312  020122     043
 3339  025315     045  043045  051117  MSG2A:  .ASCII  /%%FOR DRIVE ADDRESS TEST;/
 3340  025322  042040  044522  042526
 3341  025330  040440  042104  042522
 3342  025336  051523  052040  051505
 3343  025344  035524
 3344  025346  020045  047105  042524          .ASCII  /% ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.%/
 3345  025354  020122  054105  052120
 3346  025362  042040  044522  042526
 3347  025370  047040  046525  042502
 3348  025376  026122  040440  046114
 3349  025404  047440  044124  051105
 3350  025412  020123  044123  052517
 3351  025420  042114  041040  020105
 3352  025426  047516  026516  054105
 3353  025434  051511  040524  052116
 3354  025442  021456
 3355  025444  047045  047117  042455  MSG3:   .ASCII  /%NON-EXIST DRIVE %/
 3356  025452  044530  052123  042040
 3357  025460  044522  042526  021440
 3358  025466  051045  020110  042504  MSG4:   .ASCII  /%RH DETECTED %/
 3359  025474  042524  052103  042105
 3360  025502  021440
 3361  025504  052045  030115  020062  MSG5:   .ASCII  /%TM02 DETECTED %/
 3362  025512  042504  042524  052103
 3363  025520  042105  021440
 3364  025524  054105  052120  047055  MSG6:   .ASCII  /EXPT-NOT RECVD%/
 3365  025532  052117  051040  041505
 3366  025540  042126     043
 3367  025543     122  053103  026504  MSG7:   .ASCII  /RCVD-NOT EXPT%/
 3368  025550  047516  020124  054105
 3369  025556  052120     043
 3370  025561     045  046123  053101  MSG8:   .ASCII  /%SLAVE NUMBER %/
 3371  025566  020105  052516  041115
 3372  025574  051105  021440
 3373  025600  022445  047506  020122  MSG8A:  .ASCII  /%%FOR SLAVE ADDRESS TEST;/
 3374  025606  046123  053101  020105
 3375  025614  042101  051104  051505
 3376  025622  020123  042524  052123
 3377  025630     073
```

```
3378  025631     045  042440  052116          ,ASCII  /% ENTER EXPT SLAVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT,0/
3379  025636  051105  042440  050130
3380  025644  020124  046123  053101
3381  025652  020105  052516  041115
3382  025660  051105  020054  046101
3383  025666  020114  052117  042510
3384  025674  051522  051440  047510
3385  025702  046125  020104  042502
3386  025710  047040  047117  042455
3387  025716  044530  052123  047101
3388  025724  027124     043
3389  025727     045  047516  026516  MSG9:   ,ASCII  /%NON-EXIST SLAVE 0/
3390  025734  054105  051511  020124
3391  025742  046123  053101  020105
3392  025750     043
3393  025751     045  042522  042101  MSG10:  ,ASCII  /%READ CONT BUS PAR 0/
3394  025756  041440  047117  020124
3395  025764  052502  020123  040520
3396  025772  020122     043
3397  025775     045  051127  052111  MSG11:  ,ASCII  /%WRITE CONT BUS PAR 0/
3398  026002  020105  047503  052116
3399  026010  041040  051525  050040
3400  026016  051101  021440
3401  026022  042440  050130  020124  MSG12:  ,ASCII  / EXPT 0/
3402  026030     043
3403  026031     040  041522  042126  MSG13:  ,ASCII  / RCVD 0/
3404  026036  021440
3405  026040  046445  020122  044502  MSG14:  ,ASCII  /%MR BITS 4-0 0/
3406  026046  051524  032040  030055
3407  026054     043
3408  026055     045  051115  041040  MSG15:  ,ASCII  /%MR BITS 15-7 0/
3409  026062  052111  020123  032461
3410  026070  033455     043
3411  026073     045  052111  051105  MSG16:  ,ASCII  /%ITER: 0/
3412  026100  020072     043
3413  026103     045  041524  041040  MSG17:  ,ASCII  /%TC BIT 13 0/
3414  026110  052111  030440  020063
3415  026116     043
3416  026117     045  041524  041040  MSG18:  ,ASCII  /%TC BITS 12-0 0/
3417  026124  052111  020123  031061
3418  026132  030055  021440
3419  026136  043045  020103  044502  MSG19:  ,ASCII  /%FC BITS 15-0 0/
3420  026144  051524  030440  026465
3421  026152  020060     043
3422  026155     045  052506  020116  MSG20:  ,ASCII  /%FUN CODE BITS 5-1 OF C1 0/
3423  026162  047503  042504  041040
3424  026170  052111  020123  026465
3425  026176  020061  043117  041440
3426  026204  020061     043
3427  026207     045  047507  041040  MSG21:  ,ASCII  /%GO BIT NOT CORRECT AT START 0/
3428  026214  052111  047040  052117
3429  026222  041440  051117  042522
3430  026230  052103  040440  020124
3431  026236  052123  051101  020124
3432  026244     043
3433  026245     045  047507  041040  MSG22:  ,ASCII  /%GO BIT NOT SET 0/
```

```
3434  026252  052111  047040  052117
3435  026260  051440  052105  021440
3436  026266  043445  020117  044502  MSG23:  .ASCII  /%GO BIT NOT RESET BY INIT %/
3437  026274  020124  047516  020124
3438  026302  042522  042523  020124
3439  026310  054502  044440  044516
3440  026316  020124     043
3441  026321     045  051104  020131  MSG24:  .ASCII  /%DRY NOT SET BY INIT %/
3442  026326  047516  020124  042523
3443  026334  020124  054502  044440
3444  026342  044516  020124     043
3445  026347     045  051104  020131  MSG25:  .ASCII  /%DRY NOT RESET BY GO=1%/
3446  026354  047516  020124  042522
3447  026362  042523  020124  054502
3448  026370  043440  036517  021461
3449  026376  042045  054522  047040  MSG25A: .ASCII  /%DRY NOT SET BY GO=0%/
3450  026404  052117  051440  052105
3451  026412  041040  020131  047507
3452  026420  030075     043
3453  026423     045  047516  044440  MSG26:  .ASCII  /%NO INTERRUPT RETURNED%/
3454  026430  052116  051105  052522
3455  026436  052120  051040  052105
3456  026444  051125  042516  021504
3457  026452  041045  042101  051440  MSG27:  .ASCII  /%BAD STATUS%/
3458  026460  040524  052524  021523
3459  026466  051440  035116  021440  MSG30:  .ASCII  / SN: %/
3460  026474  042445  051122  047040  MSG31:  .ASCII  /%ERR NOT SET %/
3461  026502  052117  051440  052105
3462  026510  021440
3463  026512  040445  040524  047040  MSG32:  .ASCII  /%ATA NOT SET %/
3464  026520  052117  051440  052105
3465  026526  021440
3466  026530  040445  020123  044502  MSG33:  .ASCII  /%AS BIT NOT SET %/
3467  026536  020124  047516  020124
3468  026544  042523  020124     043
3469  026551     045  041523  047040  MSG34:  .ASCII  /%SC NOT SET %/
3470  026556  052117  051440  052105
3471  026564  021440
3472  026566  052045  042522  047040  MSG35:  .ASCII  /%TRE NOT SET %/
3473  026574  052117  051440  052105
3474  026602  021440
3475  026604  051445  040514  047040  MSG36:  .ASCII  /%SLA NOT SET %/
3476  026612  052117  051440  052105
3477  026620  021440
3478  026622  051445  041523  047040  MSG37:  .ASCII  /%SSC NOT SET %/
3479  026630  052117  051440  052105
3480  026636  021440
3481  026640  037440  021440          MSG40:  .ASCII  / ? %/
3482  026644  022445  047105  020104  MSG41:  .ASCII  /%%END OF PASS %/
3483  026652  043117  050040  051501
3484  026660  020123     043
3485  026663     045  042504  042101  MSG42:  .ASCII  /%DEAD TRACK %/
3486  026670  052040  040522  045503
3487  026676  021440
3488  026700  022445  040515  052516  MSG43:  .ASCII  /%%MANUAL TESTS (14-17) INHIBITED: HALT%/
3489  026706  046101  052040  051505
```

```
3490  026714  051524  024040  032061
3491  026722  030455  024467  044440
3492  026730  044116  041111  052111
3493  026736  042105  020072  040510
3494  026744  052114     045
3495  026747     122  051505  046105          .ASCII  /RESELECT AND PRESS CONTINUE@@/
3496  026754  041505  020124  047101
3497  026762  020104  051120  051505
3498  026770  020123  047503  052116
3499  026776  047111  042525  021445
3500  027004  051045  043505  051511  MSG44:  .ASCII  /%REGISTER START: @/
3501  027012  042524  020122  052123
3502  027020  051101  035124  021440
3503  027026  053045  041505  047524  MSG45:  .ASCII  /%VECTOR ADDRESS: @/
3504  027034  020122  042101  051104
3505  027042  051505  035123  021440
3506  027050  041445  030523  020040  MSG46:  .ASCII  /%CS1    WC     BA     FC     CS2    DS     ER     AS/
3507  027056  020040  041527  020040
3508  027064  020040  041040  020101
3509  027072  020040  020040  041506
3510  027100  020040  020040  041440
3511  027106  031123  020040  020040
3512  027114  051504  020040  020040
3513  027122  042440  020122  020040
3514  027130  020040  051501
3515  027134  020040  020040  046440          .ASCII  /     MR     TC@@/
3516  027142  020122  020040  020040
3517  027150  041524  021445
3518  027154  047045  052117  051040  MSG47:  .ASCII  /%NOT RESET BY DRIVE CLEAR@/
3519  027162  051505  052105  041040
3520  027170  020131  051104  053111
3521  027176  020105  046103  040505
3522  027204  021522
3523  027206  040445  050114  040510  MSG50:  .ASCII  /%ALPHA NOT SET@/
3524  027214  047040  052117  051440
3525  027222  052105     043
3526  027225     045  047125  054105  MSG51:  .ASCII  /%UNEXPECTED ERROR BITS@/
3527  027232  042520  052103  042105
3528  027240  042440  051122  051117
3529  027246  041040  052111  021523
3530  027254  047045  055122  047440  MSG52:  .ASCII  /%NRZ ONLY: @/
3531  027262  046116  035131  021440
3532  027270  041045  042101  046040  MSG53:  .ASCII  /%BAD LRC @/
3533  027276  041522  021440
3534  027302  041045  042101  041440  MSG54:  .ASCII  /%BAD CK @/
3535  027310  020113     043
3536  027313     045  042523  052524  MSG55:  .ASCII  /%SETUP ERROR: CHECK WRAP @ WITH TEST 50@/
3537  027320  020120  051105  047522
3538  027326  035122  041440  042510
3539  027334  045503  053440  040522
3540  027342  020120  020060  044527
3541  027350  044124  052040  051505
3542  027356  020124  030065     043
3543  027363     045  052123  052101  MSG56:  .ASCII  /%STATIC TESTS ONLY: @/
3544  027370  041511  052040  051505
3545  027376  051524  047440  046116
```

```
  3546  027404  035131  021440
```

```
3547                                                    ;TEST HEADER**********
3548
3549  027410  022445  047514  044507  MSLT1:  .ASCII  /%%LOGIC TEST 1: DRIVE ADDRESSING (M8909 RH)0/
3550  027416  020103  042524  052123
3551  027424  030440  020072  051104
3552  027432  053111  020105  042101
3553  027440  051104  051505  044523
3554  027446  043516  024040  034115
3555  027454  030071  020071  044122
3556  027462  021451
3557  027464  022445  047514  044507  MSLT2:  .ASCII  /%%LOGIC TEST 2: REGISTER ADDRESSING (M8909 RH)0/
3558  027472  020103  042524  052123
3559  027500  031040  020072  042522
3560  027506  044507  052123  051105
3561  027514  040440  042104  042522
3562  027522  051523  047111  020107
3563  027530  046450  034470  034460
3564  027536  051040  024510     043
3565  027543     045  046045  043517  MSLT3:  .ASCII  /%%LOGIC TEST 3: CONTROL BUS TEST (RH M8905 M8909)0/
3566  027550  041511  052040  051505
3567  027556  020124  035063  041440
3568  027564  047117  051124  046117
3569  027572  041040  051525  052040
3570  027600  051505  020124  051050
3571  027606  020110  034115  030071
3572  027614  020065  034115  030071
3573  027622  024471     043
3574  027625     045  046045  043517  MSLT4:  .ASCII  /%%LOGIC TEST 4: SLAVE ADDRESSING (M8905 M8903)0/
3575  027632  041511  052040  051505
3576  027640  020124  035064  051440
3577  027646  040514  042526  040440
3578  027654  042104  042522  051523
3579  027662  047111  020107  046450
3580  027670  034470  032460  046440
3581  027676  034470  031460  021451
3582  027704  022445  047514  044507  MSLT5:  .ASCII  /%%LOGIC TEST 5: MR BIT TEST (M8905)0/
3583  027712  020103  042524  052123
3584  027720  032440  020072  051115
3585  027726  041040  052111  052040
3586  027734  051505  020124  046450
3587  027742  034470  032460  021451
3588  027750  022445  047514  044507  MSLT6:  .ASCII  /%%LOGIC TEST 6: TC BIT TEST (M8905)0/
3589  027756  020103  042524  052123
3590  027764  033040  020072  041524
3591  027772  041040  052111  052040
3592  030000  051505  020124  046450
3593  030006  034470  032460  021451
3594  030014  022445  047514  044507  MSLT7:  .ASCII  /%%LOGIC TEST 7: FC BIT TEST (M8905)0/
3595  030022  020103  042524  052123
3596  030030  033440  020072  041506
3597  030036  041040  052111  052040
3598  030044  051505  020124  046450
3599  030052  034470  032460  021451
3600  030060  022445  047514  044507  MSLT10: .ASCII  /%%LOGIC TEST 10: FUNCTION BIT TEST (M8905)0/
3601  030066  020103  042524  052123
3602  030074  030440  031060  043040
```

```
  3603  030102  047125  052103  047511
  3604  030110  020116  044502  020124
  3605  030116  042524  052123  024040
  3606  030124  034115  030071  024465
  3607  030132     043
  3608  030133     045  046045  043517  MSLT11: .ASCII  /%%LOGIC TEST 11: GO BIT TEST (M8909)%/
  3609  030140  041511  052040  051505
  3610  030146  020124  030461  020072
  3611  030154  047507  041040  052111
  3612  030162  052040  051505  020124
  3613  030170  046450  034470  034460
  3614  030176  021451
  3615  030200  022445  047514  044507  MSLT12: .ASCII  /%%LOGIC TEST 12: DRIVE READY BIT (M8909)%/
  3616  030206  020103  042524  052123
  3617  030214  030440  035062  042040
  3618  030222  044522  042526  051040
  3619  030230  040505  054504  041040
  3620  030236  052111  024040  034115
  3621  030244  030071  024471     043
  3622  030251     045  046045  043517  MSLT13: .ASCII  /%%LOGIC TEST 13: INTERRUPT TEST (RH)%/
  3623  030256  041511  052040  051505
  3624  030264  020124  031461  020072
  3625  030272  047111  042524  051122
  3626  030300  050125  020124  042524
  3627  030306  052123  024040  044122
  3628  030314  021451
  3629  030316  022445  047514  044507  MSLT14: .ASCII  /%%LOGIC TEST 14: MANUAL STATUS TEST 1%/
  3630  030324  020103  042524  052123
  3631  030332  030440  035064  046440
  3632  030340  047101  040525  020114
  3633  030346  052123  052101  051525
  3634  030354  052040  051505  020124
  3635  030362  021461
  3636  030364  022445  047514  044507  MSLT15: .ASCII  /%%LOGIC TEST 15: MANUAL STATUS TEST 2%/
  3637  030372  020103  042524  052123
  3638  030400  030440  035065  046440
  3639  030406  047101  040525  020114
  3640  030414  052123  052101  051525
  3641  030422  052040  051505  020124
  3642  030430  021462
  3643  030432  022445  047514  044507  MSLT16: .ASCII  /%%LOGIC TEST 16: MANUAL STATUS TEST 3%/
  3644  030440  020103  042524  052123
  3645  030446  030440  035066  046440
  3646  030454  047101  040525  020114
  3647  030462  052123  052101  051525
  3648  030470  052040  051505  020124
  3649  030476  021463
  3650  030500  022445  047514  044507  MSLT17: .ASCII  /%%LOGIC TEST 17: MANUAL STATUS TEST 4%/
  3651  030506  020103  042524  052123
  3652  030514  030440  035067  046440
  3653  030522  047101  040525  020114
  3654  030530  052123  052101  051525
  3655  030536  052040  051505  020124
  3656  030544  021464
  3657  030546  022445  047514  044507  MSLT20: .ASCII  /%%LOGIC TEST 20: ILLEGAL FUNCTION TEST (M8909)%/
  3658  030554  020103  042524  052123
```

```
3659  030562  031040  035060  044440
3660  030570  046114  043505  046101
3661  030576  043040  047125  052103
3662  030604  047511  020116  042524
3663  030612  052123  024040  034115
3664  030620  030071  024471     043
3665  030625     045  046045  043517  MSLT21: .ASCII  /%%LOGIC TEST 21: RMR(M8909)%/
3666  030632  041511  052040  051505
3667  030640  020124  030462  020072
3668  030646  046522  024122  034115
3669  030654  030071  024471     043
3670  030661     045  046045  043517  MSLT22: .ASCII  /%%LOGIC TEST 22: CPAR(M8909)%/
3671  030666  041511  052040  051505
3672  030674  020124  031062  020072
3673  030702  050103  051101  046450
3674  030710  034470  034460  021451
3675  030716  022445  047514  044507  MSLT23: .ASCII  /%%LOGIC TEST 23: FMT(M8905 M8906)%/
3676  030724  020103  042524  052123
3677  030732  031040  035063  043040
3678  030740  052115  046450  034470
3679  030746  032460  046440  034470
3680  030754  033060  021451
3681  030760  022445  047514  044507  MSLT24: .ASCII  /%%LOGIC TEST 24: DPAR(M8906 RH)%/
3682  030766  020103  042524  052123
3683  030774  031040  035064  042040
3684  031002  040520  024122  034115
3685  031010  030071  020066  044122
3686  031016  021451
3687  031020  022445  047514  044507  MSLT25: .ASCII  /%%LOGIC TEST 25: NEF(M8909)%/
3688  031026  020103  042524  052123
3689  031034  031040  035065  047040
3690  031042  043105  046450  034470
3691  031050  034460  021451
3692  031054  022445  047514  044507  MSLT26: .ASCII  /%%LOGIC TEST 26: FCE(M8909)%/
3693  031062  020103  042524  052123
3694  031070  031040  035066  043040
3695  031076  042503  046450  034470
3696  031104  034460  021451
3697  031110  022445  047514  044507  MSLT27: .ASCII  /%%LOGIC TEST 27: ILR(M8909)%/
3698  031116  020103  042524  052123
3699  031124  031040  035067  044440
3700  031132  051114  046450  034470
3701  031140  034460  021451
3702  031144  022445  047514  044507  MSLT30: .ASCII  /%%LOGIC TEST 30:DTE(M8906 RH)%/
3703  031152  020103  042524  052123
3704  031160  031440  035060  052104
3705  031166  024105  034115  030071
3706  031174  020066  044122  021451
3707  031202  022445  047514  044507  MSLT31: .ASCII  /%%LOGIC TEST 31: OPI(M8903)%/
3708  031210  020103  042524  052123
3709  031216  031440  035061  047440
3710  031224  044520  046450  034470
3711  031232  031460  021451
3712  031236  022445  047514  044507  MSLT32: .ASCII  /%%LOGIC TEST 32: UNS(M8909)%/
3713  031244  020103  042524  052123
3714  031252  031440  035062  052440
```

```
  3715  031260  051516  046450  034470
  3716  031266  034460  021451
  3717  031272  022445  047514  044507  MSLT33: ,ASCII  /%%LOGIC TEST 33: PIP(M8909)%/
  3718  031300  020103  042524  052123
  3719  031306  031440  035063  050040
  3720  031314  050111  046450  034470
  3721  031322  034460  021451
  3722  031326  022445  047514  044507  MSLT34: ,ASCII  /%%LOGIC TEST 34: PES(M8911)%/
  3723  031334  020103  042524  052123
  3724  031342  031440  035064  050040
  3725  031350  051505  046450  034470
  3726  031356  030461  021451
  3727  031362  022445  047514  044507  MSLT35: ,ASCII  /%%LOGIC TEST 35: TCW(M8903 M8905)%/
  3728  031370  020103  042524  052123
  3729  031376  031440  035065  052040
  3730  031404  053503  046450  034470
  3731  031412  031460  046440  034470
  3732  031420  032460  021451
  3733  031424  022445  047514  044507  MSLT36: ,ASCII  /%%LOGIC TEST 36: FCS(M8903 M8905)%/
  3734  031432  020103  042524  052123
  3735  031440  031440  035066  043040
  3736  031446  051503  046450  034470
  3737  031454  031460  046440  034470
  3738  031462  032460  021451
  3739  031466  022445  047514  044507  MSLT37: ,ASCII  /%%LOGIC TEST 37: ACCL(M8903 M8905)%/
  3740  031474  020103  042524  052123
  3741  031502  031440  035067  040440
  3742  031510  041503  024114  034115
  3743  031516  030071  020063  034115
  3744  031524  030071  024465     043
  3745  031531     045  046045  043517  MSLT40: ,ASCII  /%%LOGIC TEST 40: PE TAPE MARK(M8902)%/
  3746  031536  041511  052040  051505
  3747  031544  020124  030064  020072
  3748  031552  042520  052040  050101
  3749  031560  020105  040515  045522
  3750  031566  046450  034470  031060
  3751  031574  021451
  3752  031576  022445  047514  044507  MSLT41: ,ASCII  /%%LOGIC TEST 41: NRZ TAPE MARK (M8904)%/
  3753  031604  020103  042524  052123
  3754  031612  032040  035061  047040
  3755  031620  055122  052040  050101
  3756  031626  020105  040515  045522
  3757  031634  024040  034115  030071
  3758  031642  024464     043
  3759  031645     045  046045  043517  MSLT42: ,ASCII  /%%LOGIC TEST 42: WRAP 3,NRZ,NORMAL,ODD%/
  3760  031652  041511  052040  051505
  3761  031660  020124  031064  020072
  3762  031666  051127  050101  031440
  3763  031674  047054  055122  047054
  3764  031702  051117  040515  026114
  3765  031710  042117  021504
  3766  031714  022445  047514  044507  MSLT43: ,ASCII  /%%LOGIC TEST 43: WRAP 3,PE,NORMAL,ODD%/
  3767  031722  020103  042524  052123
  3768  031730  032040  035063  053440
  3769  031736  040522  020120  026063
  3770  031744  042520  047054  051117
```

```
 3771  031752  040515  026114  042117
 3772  031760  021504
 3773  031762  022445  047514  044507  MSLT44: .ASCII  /%%LOGIC TEST 44: WRAP 2,NRZ,NORMAL,ODD%/
 3774  031770  020103  042524  052123
 3775  031776  032040  035064  053440
 3776  032004  040522  020120  026062
 3777  032012  051116  026132  047516
 3778  032020  046522  046101  047454
 3779  032026  042104     043
 3780  032031     045  046045  043517  MSLT45: .ASCII  /%%LOGIC TEST 45: WRAP 2,PE,NORMAL,ODD%/
 3781  032036  041511  052040  051505
 3782  032044  020124  032464  020072
 3783  032052  051127  050101  031040
 3784  032060  050054  026105  047516
 3785  032066  046522  046101  047454
 3786  032074  042104     043
 3787  032077     045  046045  043517  MSLT46: .ASCII  /%%LOGIC TEST 46: WRAP 1,NRZ,NORMAL,ODD%/
 3788  032104  041511  052040  051505
 3789  032112  020124  033064  020072
 3790  032120  051127  050101  030440
 3791  032126  047054  055122  047054
 3792  032134  051117  040515  026114
 3793  032142  042117  021504
 3794  032146  022445  047514  044507  MSLT47: .ASCII  /%%LOGIC TEST 47: WRAP 1,PE,NORMAL,ODD%/
 3795  032154  020103  042524  052123
 3796  032162  032040  035067  053440
 3797  032170  040522  020120  026061
 3798  032176  042520  047054  051117
 3799  032204  040515  026114  042117
 3800  032212  021504
 3801  032214  022445  047514  044507  MSLT50: .ASCII  /%%LOGIC TEST 50: WRAP 0,NRZ,NORMAL,ODD%/
 3802  032222  020103  042524  052123
 3803  032230  032440  035060  053440
 3804  032236  040522  020120  026060
 3805  032244  051116  026132  047516
 3806  032252  046522  046101  047454
 3807  032260  042104     043
 3808  032263     045  046045  043517  MSLT51: .ASCII  /%%LOGIC TEST 51: WRAP 0,PE,NORMAL,ODD%/
 3809  032270  041511  052040  051505
 3810  032276  020124  030465  020072
 3811  032304  051127  050101  030040
 3812  032312  050054  026105  047516
 3813  032320  046522  046101  047454
 3814  032326  042104     043
 3815  032331     045  046045  043517  MSLT52: .ASCII  /%%LOGIC TEST 52: CORE DUMP WRITE (M8906)%/
 3816  032336  041511  052040  051505
 3817  032344  020124  031065  020072
 3818  032352  047503  042522  042040
 3819  032360  046525  020120  051127
 3820  032366  052111  020105  046450
 3821  032374  034470  033060  021451
 3822  032402  022445  047514  044507  MSLT53: .ASCII  /%%LOGIC TEST 53: CORE DUMP READ (M8906)%/
 3823  032410  020103  042524  052123
 3824  032416  032440  035063  041440
 3825  032424  051117  020105  052504
 3826  032432  050115  041040  040505
```

```
 3827  032440  020104  046450  034470
 3828  032446  033060  021451
 3829  032452  022445  047514  044507  MSLT54: ,ASCII  /%%LOGIC TEST 54: EVEN PARITY WRITE (M8903 M8904)%/
 3830  032460  020103  042524  052123
 3831  032466  032440  035064  042440
 3832  032474  042526  020116  040520
 3833  032502  044522  054524  053440
 3834  032510  044522  042524  024040
 3835  032516  034115  030071  020063
 3836  032524  034115  030071  024464
 3837  032532     043
 3838  032533     045  046045  043517  MSLT55: ,ASCII  /%%LOGIC TEST 55: EVEN PARITY READ(M8903 M8904)%/
 3839  032540  041511  052040  051505
 3840  032546  020124  032465  020072
 3841  032554  053105  047105  050040
 3842  032562  051101  052111  020131
 3843  032570  042522  042101  046450
 3844  032576  034470  031460  046440
 3845  032604  034470  032060  021451
 3846  032612  022445  047514  044507  MSLT56: ,ASCII  /%%LOGIC TEST 56: READ REVERSE(M8906)%/
 3847  032620  020103  042524  052123
 3848  032626  032440  035066  051040
 3849  032634  040505  020104  042522
 3850  032642  042526  051522  024105
 3851  032650  034115  030071  024466
 3852  032656     043
 3853  032657     045  046045  043517  MSLT57: ,ASCII  /%%LOGIC TEST 57: CRC(M8904)%/
 3854  032664  041511  052040  051505
 3855  032672  020124  033465  020072
 3856  032700  051103  024103  034115
 3857  032706  030071  024464     043
 3858  032713     045  046045  043517  MSLT60: ,ASCII  /%%LOGIC TEST 60: LRC(M8904)%/
 3859  032720  041511  052040  051505
 3860  032726  020124  030066  020072
 3861  032734  051114  024103  034115
 3862  032742  030071  024464     043
 3863  032747     045  046045  043517  MSLT61: ,ASCII  /%%LOGIC TEST 61: CORRECTABLE DATA (M8902 M8901)%/
 3864  032754  041511  052040  051505
 3865  032762  020124  030466  020072
 3866  032770  047503  051122  041505
 3867  032776  040524  046102  020105
 3868  033004  040504  040524  024040
 3869  033012  034115  030071  020062
 3870  033020  034115  030071  024461
 3871  033026     043
 3872  033027     045  046045  043517  MSLT62: ,ASCII  /%%LOGIC TEST 62: INCORRECTABLE DATA (M8902 M8904)%/
 3873  033034  041511  052040  051505
 3874  033042  020124  031066  020072
 3875  033050  047111  047503  051122
 3876  033056  041505  040524  046102
 3877  033064  020105  040504  040524
 3878  033072  024040  034115  030071
 3879  033100  020062  034115  030071
 3880  033106  024464     043
 3881  033111     045  046045  043517  MSLT63: ,ASCII  /%%LOGIC TEST 63: PEF(M8902)%/
 3882  033116  041511  052040  051505
```

```
  3883  033124  020124  031466  020072
  3884  033132  042520  024106  034115
  3885  033140  030071  024462     043
  3886  033145     045  046045  043517  MSLT64: .ASCII  /%%LOGIC TEST 64: FC OVERFLOW (M8985)#/
  3887  033152  041511  052040  051505
  3888  033160  020124  032066  020072
  3889  033166  041506  047440  042526
  3890  033174  043122  047514  020127
  3891  033202  046450  034470  032460
  3892  033210  021451
```

```
                                                ;MANUAL INSTRUCTION**********

 033212  052045  050131  020105  MMSG0:  .ASCII  /%TYPE CR WHEN READY;0/
 033220  051103  053440  042510
 033226  020116  042522  042101
 033234  035531     043
 033237     045  046445  052517  MMSG1:  .ASCII  /%%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE;0/
 033244  052116  052040  050101
 033252  020105  044527  044124
 033260  047040  020117  051127
 033266  052111  020105  044522
 033274  043516  020054  047514
 033302  042101  052040  020117
 033310  047502  026124  051440
 033316  052105  052040  020117
 033324  047117  046040  047111
 033332  035105     043
 033335     045  042523  020124  MMSG2:  .ASCII  /%SET TO OFFLINE;0/
 033342  047524  047440  043106
 033350  044514  042516  021472
 033356  046445  053117  020105  MMSG3:  .ASCII  /%MOVE FORWARD TO EOT, ONLINE;0/
 033364  047506  053522  051101
 033372  020104  047524  042440
 033400  052117  020054  047117
 033406  044514  042516  021472
 033414  047445  043106  046040  MMSG4:  .ASCII  /%OFF LINE REVERSE PAST EOT, INSERT WRITE RING, ON LINE0/
 033422  047111  020105  042522
 033430  042526  051522  020105
 033436  040520  052123  042440
 033444  052117  020054  047111
 033452  042523  052122  053440
 033460  044522  042524  051040
 033466  047111  026107  047440
 033474  020116  044514  042516
 033502     043
 033503     045  046445  053117  MMSG5:  .ASCII  /%%MOVE TAPE TO BOT; ON LINE0/
 033510  020105  040524  042520
 033516  052040  020117  047502
 033524  035524  047440  020116
 033532  044514  042516     043
```

```
                                                ;TAG MESSAGE
033537      045  046123  020101  TM811:   .ASCII  /%SLA %/
033544      043
033545      045  047502  020124  TM812:   .ASCII  /%BOT %/
033552      043
033553      045  046524  021440  TM813:   .ASCII  /%TM %/
033560  044445  041104  021440  TM814:   .ASCII  /%IDB %/
033566  051445  053504  020116  TM815:   .ASCII  /%SDWN %/
033574      043
033575      045  042520  020123  TM816:   .ASCII  /%PES %/
033602      043
033603      045  051523  020103  TM817:   .ASCII  /%SSC %/
033610      043
033611      045  051104  020131  TM818:   .ASCII  /%DRY %/
033616      043
033617      045  050104  020122  TM819:   .ASCII  /%DPR %/
033624      043
033625      045  052116  020114  TM8110:  .ASCII  /%NTL %/
033632      043
033633      045  047505  020124  TM8111:  .ASCII  /%EOT %/
033640      043
033641      045  051127  020114  TM8112:  .ASCII  /%WRL %/
033646      043
033647      045  047515  020114  TM8113:  .ASCII  /%MOL %/
033654      043
033655      045  044520  020120  TM8114:  .ASCII  /%PIP %/
033662      043
033663      045  051105  020122  TM8115:  .ASCII  /%ERR %/
033670      043
033671      045  052101  020101  TM8116:  .ASCII  /%ATA %/
033676      043
033677      045  046111  020106  TM8117:  .ASCII  /%ILF %/
033704      043
033705      045  046111  020122  TM8118:  .ASCII  /%ILR %/
033712      043
033713      045  046522  020122  TM8119:  .ASCII  /%RMR %/
033720      043
033721      045  050103  051101  TM8120:  .ASCII  /%CPAR %/
033726  021440
033730  043045  052115  021440  TM8121:  .ASCII  /%FMT %/
033736  042045  040520  020122  TM8122:  .ASCII  /%DPAR %/
033744      043
033745      045  047111  020103  TM8123:  .ASCII  /%INC %/
033752      043
033753      045  050126  020105  TM8124:  .ASCII  /%VPE %/
033760      043
033761      045  042520  020106  TM8125:  .ASCII  /%PEF %/
033766      043
033767      045  051114  020103  TM8126:  .ASCII  /%LRC %/
033774      043
033775      045  051516  020107  TM8127:  .ASCII  /%NSG %/
034002      043
034003      045  041506  020105  TM8128:  .ASCII  /%FCE %/
034010      043
```

```
 3990  034011     045  051503  021440  TM829:  .ASCII  /%CS %/
 3991  034016  044445  046524  021440  TM830:  .ASCII  /%ITM %/
 3992  034024  047045  043105  021440  TM831:  .ASCII  /%NEF %/
 3993  034032  042045  042524  021440  TM832:  .ASCII  /%DTE %/
 3994  034040  047445  044520  021440  TM833:  .ASCII  /%OPI %/
 3995  034046  053445  044522  042524  TM833A: .ASCII  /%WRITE OPI %/
 3996  034054  047440  044520  021440
 3997  034062  051045  040505  020104  TM833B: .ASCII  /%READ OPI %/
 3998  034070  050117  020111     043
 3999  034075     045  047125  020123  TM834:  .ASCII  /%UNS %/
 4000  034102     043
 4001  034103     045  047503  051122  TM835:  .ASCII  /%CORR %/
 4002  034110  021440
 4003  034112  041445  041522  021440  TM836:  .ASCII  /%CRC %/
 4004  034120  052045  053503  021440  TM837:  .ASCII  /%TCW %/
 4005  034126  043045  051503  021440  TM838:  .ASCII  /%FCS %/
 4006  034134  040445  041503  020114  TM839:  .ASCII  /%ACCL %/
 4007  034142     043
 4008
 4009          034144                          .EVEN
 4010                                          ;WRITE BUFFER
 4011
 4012  034144                          WDAT%:  .REPT   100
 4013                                          -1
 4014                                          .ENDR
 4015
 4016
 4017                                          ;READ BUFFER
 4018
 4019  034344                          RDATA:  .REPT   100
 4020                                          0
 4021                                          .ENDR
 4022
 4023                                          ;WRAP AROUND MESSAGES******
 4024
 4025  034544  051445  052105  050125  WMSG2:  .ASCII  /%SETUP ERROR%%/
 4026  034552  042440  051122  051117
 4027  034560  021445
 4028  034562  050045  052101  047122  WMSG3:  .ASCII  /%PATRN NUMBER = %/
 4029  034570  047040  046525  042502
 4030  034576  020122  020075     043
 4031  034603     045  047516  026516  WMSG4:  .ASCII  /%NON-EXISTANT DRIVE%%/
 4032  034610  054105  051511  040524
 4033  034616  052116  042040  044522
 4034  034624  042526  021445
 4035  034630  041445  030523  021440  WMSG6:  .ASCII  /%CS1 %/
 4036  034636  053445  020103     043  WMSG6A: .ASCII  /%WC %/
 4037  034643     045  040502  021440  WMSG6B: .ASCII  /%BA %/
 4038  034650  043045  020103     043  WMSG6C: .ASCII  /%FC %/
 4039  034655     045  051503  020062  WMSG6D: .ASCII  /%CS2 %/
 4040  034662     043
 4041  034663     045  051504  021440  WMSG6E: .ASCII  /%DB %/
 4042  034670  042445  020122     043  WMSG6F: .ASCII  /%ER %/
 4043  034675     045  051501  021440  WMSG6G: .ASCII  /%AS %/
 4044  034702  041445  020103     043  WMSG6H: .ASCII  /%CC %/
 4045  034707     045  041104  021440  WMSG6I: .ASCII  /%DR %/
```

```
 4046  034714  046445  020122     043  WMSG6J: .ASCII  /%MR %/
 4047  034721     045  052104  021440  WMSG6K: .ASCII  /%DT %/
 4048  034726  052045  020103     043  WMSG6L: .ASCII  /%TC %/
 4049  034733     045  047123  021440  WMSG6M: .ASCII  /%SN %/
 4050  034740  041045  042101  042040  WMSG16: .ASCII  /%BAD DATA%/
 4051  034746  052101  021501
 4052  034752  043445  020072     043  WMSG17: .ASCII  /%G: %/
 4053  034757     045  035102  021440  WMSG20: .ASCII  /%B: %/
 4054  034764  041445  035116  021440  WMSG21: .ASCII  /%CN: %/
 4055  034772  041045  042101  051440  WMSG23: .ASCII  /%BAD STATUS%/
 4056  035000  040524  052524  021523
 4057  035006  047045  020117  047111  WMSG24: .ASCII  /%NO INTERRUPT%/
 4058  035014  042524  051122  050125
 4059  035022  021524
 4060  035024  047045  020117  046103  WMSG25: .ASCII  /%NO CLOCK UP%/
 4061  035032  041517  020113  050125
 4062  035040     043
 4063  035041     045  047516  041440  WMSG26: .ASCII  /%NO CLOCK DOWN%/
 4064  035046  047514  045503  042040
 4065  035054  053517  021516
 4066  035060  042045  052101  020101  WMSG27: .ASCII  /%DATA PAT:%/
 4067  035066  040520  035124     043
 4068  035073     045  040502  020104  WMSG28: .ASCII  /%BAD PREAMBLE%/
 4069  035100  051120  040505  041115
 4070  035106  042514     043
 4071  035111     045  040502  020104  WMSG29: .ASCII  /%BAD POSTAMBLE%/
 4072  035116  047520  052123  046501
 4073  035124  046102  021505
 4074  035130  042445  051117  041440  WMSG31: .ASCII  /%EOR CLEAR DID NOT CLEAR GO%%/
 4075  035136  042514  051101  042040
 4076  035144  042111  047040  052117
 4077  035152  041440  042514  051101
 4078  035160  043440  022517     043
 4079  035165     040  040520  051124  WMSG32: .ASCII  / PATRN %/
 4080  035172  020116     043
 4081
 4082  035175     045  043536     043  $CNTG:  .ASCII  /%^G%/
 4083  035201     045  053523  036522  $MSWR:  .ASCII  /%SWR= %/
 4084  035206  021440
 4085  035210  020040  042516  036527  $MNEW:  .ASCII  /  NEW= %/
 4086  035216  021440
 4087  035220  037440  022440     043  $QUEST: .ASCII  / ? %%/
 4088
 4089          035226                  PRL:    .EVEN
 4090  035226  000000                          0
 4091                                          .REPT   50
 4092                                          0
 4093                                          .ENDR
 4094  035350  000000                  POST:   0
 4095                                          .REPT   50
 4096                                          0
 4097                                          .ENDR
 4098  035472  000000                  WBUFF:  0
 4099          036104                          .=.+410
 4100  036104  000000                  RBUFF:  0
 4101
```

```
  4102          000001                      .END
```

```
ADDFL   000746
AS      000526
ASF     000730
ATAF    000720
ATTA    022746
ATTB    022770
ATTC    023012
ATTD    023022
ATTN    022726
ATTP    023032
ATTPA   023060
ATTX    023104
ATTXA   023116
ATTXX   023140
BA      000514
BOTT    016620
BOTTA   016644
BOTTX   016660
CC      000530
CKSWR   024442
CNTLU   024514
COUNT   024436
CRCNT   000774
CS      000520
C1      000510
DATAD   000760
DATA0   001032
DATA1   001034
DATA2   001036
DATA3   001040
DATBL   001030
DATC    000754
DAT1    015400
DAT1A   015404
DAT1B   015410
DAT2    015422
DAT3    015430
DAT4    015440
DAT4A   015446
DB      000532
DCA     022514
DCA0    022530
DCB     022570
DCC     022600
DCD     022470
DCE     022610
DCEA    022636
DCEX    022656
DCEXA   022670
DCEXX   022712
DCHK    017416
DCHKA   017472
DCHKA0  017446
DCHKA1  017522
DCHKA2  017550
DCHKB   017622
DCHKC   017672
DCHKD   017700
DCHKE   020250
DCHKX   020204
DCHKX1  020226
DCHK0   017702
DCHK1   017734
DCHK1A  017772
DCHK2   020062
DCHK2A  020126
DCHK2B  020130
DCHK3   020154
DCKEP   020606
DCKEP0  020636
DCKE0   020264
DCKE1   020274
DCKE10  020546
DCKE11  020572
DCKE12  020602
DCKE2   020304
DCKE3   020314
DCKE4   020334
DCKE5   020346
DCKE6   020404
DCKE7   020442
DERFL   000766
DOUT    024752
DOUTD   025034
DOUT1   024766
DOUT2   025014
DOUT3   025022
DRVCLR  022464
DRVN    000624
DRVTP   000604
DS      000522
DSUP    015214
DS0     015304
DS1     015330
DS2     015350
DS3     015360
DS4     015370
DT      000536
EMADDR  000622
EORP    021602
EORPA   021576
EORPX   021766
EORP1   021636
EORP1A  021704
EORP2   021730
EORP3   021752
ER      000524
ERADD   000672
ERRF    000726
EXEC    016244
EXFL    000716
EXW2    016662
EXW2A   016710
EXW2B   016726
EXW2C   016742
EXW2E   016762
EXW2F   016770
EXW2G   017020
EXW2H   017034
EXW2J   017046
EXW2J0  017066
EXW2J1  017070
EXW2K   017110
EXW2X   017124
EX0     016262
EX1     016276
EX1A    016312
EX2     016332
EX3     016356
EX5     016410
EX5A    016414
EX5A1   016450
EX5B    016504
EX5C    016534
EX5C0   016600
EX5C1   016604
EX5D    016610
FC      000516
FUN     000752
HDRFL   000620
ILFT    000544
INIT1   023502
INIT2   023510
INMT    023462
INST    023526
ITAMT   000606
ITCNT   000702
ITER    023420
ITER0   023430
ITER1   023436
ITRLP   000714
LTADD   000742
LTGA    022034
LTGA1   022052
LTGA2   022104
LTGB    022132
LTGC    022136
LTGC0   022224
LTGD    022230
LTGER   022014
LTGER0  022006
LTGER1  023154
LTGER2  023146
LTGER3  021774
LTGX    022260
LTGXA   022272
LTGXX   022314
LTG1A   023174
LTG1B   023212
LTG1C   023322
LTG1T   023272
LTG1X   023326
LTG1XX  023362
LTG1X1  023340
LT1     002310
LT1A    002412
LT1B    002452
LT1C    002464
LT1ER   002474
LT1ER1  002504
LT1ER2  002512
LT1G    002334
LT1G0   002324
LT1X    002534
LT10    004150
LT10A   004166
LT10A1  004164
LT10B   004226
LT10E1  004250
LT10IT  004152
LT10X   004274
LT11    004304
LT11B   004336
LT11C   004402
LT11E1  004424
LT11E2  004456
LT11E3  004510
LT11IT  004306
LT11X   004536
LT12    004546
LT12B   004576
LT12C   004640
LT12E1  004660
LT12E2  004702
LT12E3  004724
LT12IT  004550
LT12X   004744
LT13    004754
LT13A   005014
LT13E1  005020
LT13IT  004764
LT13X   005046
LT14    005056
LT14A   005100
LT14IT  005116
LT14X   005156
LT14XX  005162
LT15    005166
LT15A   005210
LT15IT  005226
LT15X   005266
LT15XX  005272
LT16    005276
LT16A   005320
LT16IT  005336
LT16X   005376
LT16XX  005402
LT17    005406
LT17A   005430
LT17IT  005446
LT17X   005506
LT17XX  005512
LT2     002540
LT2A    002600
LT2B    002620
LT2C    002634
LT2ERG  002714
LT2ER1  002644
LT2ER2  002662
LT2ER3  002700
LT2IT   002542
LT2LP   002730
LT2X    002736
LT20    005516
LT20A   005544
LT20B   005622
LT20C   005632
LT20IT  005532
LT20X   005646
LT21    005662
LT21A   005766
LT21B   005776
LT21IT  005676
LT21XA  006006
LT22    006026
LT22A   006126
LT22IT  006042
LT22X   006136
LT23    006152
LT23A   006252
LT23IT  006166
LT23X   006262
LT24    006302
LT24A   006414
LT24B   006430
```

```
LT24B0  006452
LT24C   006464
LT24D   006546
LT24IT  006316
LT24X   006572
LT25    006626
LT25A   006734
LT25IT  006642
LT25X   006744
LT26    006760
LT26A   007044
LT26B   007070
LT26C   007144
LT26IT  006766
LT26W   006774
LT26X   007160
LT27    007174
LT27A   007230
LT27B   007266
LT27IT  007220
LT27X   007276
LT27XX  007306
LT3     002746
LT3A    002766
LT3B    003006
LT3C    003022
LT3ER1  003034
LT3ER2  003062
LT3IT   002750
LT3X    003104
LT3XX   003122
LT30    007312
LT30A   007414
LT30B   007456
LT30C   007516
LT30D   007534
LT30E   007564
LT30IT  007334
LT30X   007610
LT31    007630
LT31A   007724
LT31B   007766
LT31C   010004
LT31D   010052
LT31E   010122
LT31IT  007644
LT31W   007660
LT31W1  010012
LT31X   010136
LT32    010160
LT32A   010266
LT32IT  010174
LT32X   010276
LT33    010312
LT33IT  010326
LT33X   010404
LT34    010414
LT34A   010454
LT34A1  010442
LT34B   010504
LT34C   010516
LT34IT  010436
LT34X   010546
LT34XX  010552
LT35    010556
LT35A   010660
LT35IT  010572
LT35X   010720
LT36    010730
LT36A   011000
LT36IT  010744
LT36X   011040
LT37    011050
LT37A   011124
LT37B   011156
LT37IT  011064
LT37X   011210
LT4     003132
LT4A    003226
LT4B    003272
LT4C    003304
LT4D    003324
LT4ERG  003354
LT4ER1  003336
LT4ER2  003346
LT4G    003156
LT4G0   003146
LT4X    003404
LT40    011220
LT40A   011302
LT40IT  011242
LT40W   011250
LT40X   011336
LT40XX  011342
LT41    011346
LT41A   011424
LT41B   011460
LT41C   011510
LT41D   011542
LT41IT  011362
LT41X   011562
LT42    011614
LT42A   011660
LT42B   011672
LT43    011726
LT43A   011740
LT44    012010
LT44A   012054
LT44B   012066
LT45    012116
LT45A   012130
LT46    012200
LT46A   012244
LT46B   012256
LT47    012306
LT47A   012320
LT5     003410
LT5A    003430
LT5B    003454
LT5C    003464
LT5D    003474
LT5E    003520
LT5ER1  003532
LT5ER2  003556
LT5IT   003416
LT5X    003602
LT50    012374
LT50A   012440
LT50B   012452
LT51    012502
LT51A   012514
LT52    012564
LT52A   012642
LT52X   012666
LT53    012676
LT53A   012762
LT53X   013014
LT54    013024
LT55    013074
LT56    013144
LT57    013222
LT57A   013324
LT57B   013340
LT57B1  013366
LT57B2  013406
LT57C   013460
LT57D   013474
LT57E   013524
LT57IT  013252
LT57P8  013232
LT57X   013544
LT6     003612
LT6A    003630
LT6A1   003626
LT6B    003634
LT6C    003660
LT6D    003674
LT6ER1  003716
LT6ER2  003742
LT6ER4  004004
LT6IT   003614
LT6X    004014
LT60    013560
LT60C   013652
LT60D   013666
LT60E   013720
LT60F   013752
LT60IT  013600
LT60X   013772
LT61    014006
LT61A   014100
LT61A1  014112
LT61B   014122
LT61C   014136
LT61D   014164
LT61E   014166
LT61F   014230
LT61IT  014034
LT61X   014250
LT61XX  014260
LT62    014264
LT62A   014400
LT62B   014414
LT62D   014444
LT62E   014356
LT62E1  014370
LT62F   014476
LT62IT  014312
LT62X   014522
LT62XX  014532
LT63    014536
LT63A   014626
LT63B   014676
LT63C   014712
LT63D   014744
LT63IT  014564
LT63X   014764
LT63XX  014774
LT64    015000
LT64A   015070
LT64B   015102
LT64C   015106
LT64IT  015014
LT64X   015160
LT7     004024
LT7A    004042
LT7B    004066
LT7C    004032
LT7ER1  004110
LT7IT   004026
LT7X    004134
MM8G0   033212
MM8G1   033237
MM8G2   033335
MM8G3   033356
MM8G4   033414
MM8G5   033503
MR      000534
M8G1    025172
M8G10   025751
M8G11   025775
M8G12   026022
M8G13   026031
M8G14   026040
M8G15   026055
M8G16   026073
M8G17   026103
M8G18   026117
M8G19   026136
M8G2    025276
M8G2A   025315
M8G20   026155
M8G21   026207
M8G22   026245
M8G23   026266
M8G24   026321
M8G25   026347
M8G25A  026376
M8G26   026423
M8G27   026452
M8G3    025444
M8G30   026466
M8G31   026474
M8G32   026512
M8G33   026530
M8G34   026551
M8G35   026566
M8G36   026604
M8G37   026622
M8G4    025466
M8G40   026640
M8G41   026644
M8G42   026663
M8G43   026700
M8G44   027004
M8G45   027026
M8G46   027050
M8G47   027154
M8G5    025504
M8G50   027200
M8G51   027225
M8G52   027254
M8G53   027270
M8G54   027302
M8G55   027313
```

```
MSG56   027363
MSG6    025524
MSG7    025543
MSG8    025561
MSG8A   025600
MSG9    025727
MSLT1   027410
MSLT10  030060
MSLT11  030133
MSLT12  030200
MSLT13  030251
MSLT14  030316
MSLT15  030364
MSLT16  030432
MSLT17  030500
MSLT2   027464
MSLT20  030546
MSLT21  030625
MSLT22  030661
MSLT23  030716
MSLT24  030760
MSLT25  031020
MSLT26  031054
MSLT27  031110
MSLT3   027543
MSLT30  031144
MSLT31  031202
MSLT32  031236
MSLT33  031272
MSLT34  031326
MSLT35  031362
MSLT36  031424
MSLT37  031466
MSLT4   027625
MSLT40  031531
MSLT41  031576
MSLT42  031645
MSLT43  031714
MSLT44  031762
MSLT45  032031
MSLT46  032077
MSLT47  032146
MSLT5   027704
MSLT50  032214
MSLT51  032263
MSLT52  032331
MSLT53  032402
MSLT54  032452
MSLT55  032533
MSLT56  032612
MSLT57  032657
MSLT6   027750
MSLT60  032713
MSLT61  032747
MSLT62  033027
MSLT63  033111
MSLT64  033145
MSLT7   030014
MTINT   023570
NRZOF   000662
OCTP    024206
OCTPE   024176
OCTPE1  024212
OCTPG   024370
OCTPG0  024406
OCTPG1  024412
OCTP0   024234
OCTP1   024256
OCTP2   024264
OCTP3   024354
OFL     024432
OUT     024550
PATRN   001002
PC      =%000007
PCNTR   001016
PD0     020712
PD1     020726
PD2     020770
PD3     020774
PD4     021004
PD5     021124
PD6     021150
PEXFL   000736
PFLG    000666
POST    035350
PPGEN   021526
PPG0    021540
PPG1    021564
PRCHK   020674
PRE     035226
PREFL   000770
PSCHK   020644
PSW     000566
RBUFF   036104
RCDP    001010
RDAD    000762
RDATA   034344
RDRVF   001006
RDSW    024440
REGP    022332
REGS    000612
RTRN    000670
R0      =%000000
R1      =%000001
R2      =%000002
R3      =%000003
R4      =%000004
R5      =%000005
SAV1    000704
SAV2    000706
SAV3    000710
SCF     000732
SCOLP   000712
SCOPE   023374
SCOPE1  023412
SERFL   000772
SERNUM  000602
SETUP   015712
SET0    015726
SET1    015732
SET2    015766
SKAT    001014
SLAF    000722
SLVN    000664
SN      000540
SNPG    025136
SNPT    025060
SP      =%000006
SP0     016046
SP01    016032
SP1     016052
SP1A    016116
SP1B    016136
SP3     016206
SP4     016226
SP5     016242
SSCF    000724
START   001414
STATC   001012
STATF   001004
STATIC  015170
STATX   015212
STFLG   000740
STOP    025162
STSCD   002164
ST0     001672
ST1     001714
ST2     002012
SUSWR   001426
SWR     000570
SWREG   000176
TADX    001412
TC      000542
TCRLF   024076
TCRLFA  024114
TEMPST  024434
TEMP1   000674
TEMP2   000676
TEMP3   000700
TEND    002236
TENDX   002272
TEX     024174
TIB     000616
TIMER   023754
TKB     000574
TKS     000572
TM81    033537
TM810   033625
TM811   033633
TM812   033641
TM813   033647
TM814   033655
TM815   033663
TM816   033671
TM817   033677
TM818   033705
TM819   033713
TM82    033545
TM820   033721
TM821   033730
TM822   033736
TM823   033745
TM824   033753
TM825   033761
TM826   033767
TM827   033775
TM828   034003
TM829   034011
TM83    033553
TM830   034016
TM831   034024
TM832   034032
TM833   034040
TM833A  034046
TM833B  034062
TM834   034075
TM835   034103
TM836   034112
TM837   034120
TM838   034126
TM839   034134
TM84    033560
TM85    033566
TM86    033575
TM87    033603
TM88    033611
TM89    033617
TOB     000614
TOG     024160
TPB     000600
TPS     000576
TREF    000734
TR00    000626
TR01    000630
TR02    000632
TR03    000634
TR04    000636
TR05    000640
TR06    000642
TR07    000644
TR10    000646
TR11    000650
TR12    000652
TR13    000654
TR14    000656
TR15    000660
TSCD    002016
TSCD0   002050
TSCD1   002104
TSCD2   002116
TSCD3   002142
TSTTBL  001066
TTIN    023772
TTINT   023602
TTIN1   024012
TTIN2   024026
TTOUT   024044
TTR     023606
TTR0    023614
TTR1    023642
TTR1A   023656
TTR1B   023672
TTR2    023730
TTR3    023740
TTR4    023750
TTR5    023752
T1AD    001072
T1IAD   001074
T10AD   001126
T10IAD  001130
T11AD   001132
T11IAD  001134
T12AD   001136
T12IAD  001140
T13AD   001142
T13IAD  001144
T14AD   001146
T14IAD  001150
T15AD   001152
T15IAD  001154
T16AD   001156
T16IAD  001160
T17AD   001162
T17IAD  001164
T2AD    001076
```

```
T2IAD   001180
T2OAD   001166
T2OIAD  001170
T2IAD   001172
T2IIAD  001174
T2AD    001176
T2AIAD  001200
T3AD    001202
T3IAD   001204
T4AD    001206
T4FL    000744
T4IAD   001210
T5AD    001212
T5IAD   001214
T6AD    001216
T6IAD   001220
T7AD    001222
T7IAD   001224
T3AD    001102
T3IAD   001104
T3OAD   001226
T3OIAD  001230
T3IAD   001232
T3IIAD  001234
T32AD   001236
T32IAD  001240
T33AD   001242
T33IAD  001244
T34AD   001246
T34IAD  001250
T35AD   001252
T35IAD  001254
T36AD   001256
T36IAD  001260
T37AD   001262
T37IAD  001264
T4AD    001106
T4IAD   001110
T40AD   001266
T40IAD  001270
T41AD   001272
T41IAD  001274
T42AD   001276
T43AD   001302
T44AD   001306
T45AD   001312
T46AD   001316
T47AD   001322
T5AD    001112
T5IAD   001114
T50AD   001326
T51AD   001332
T52AD   001336
T53AD   001342
T54AD   001346
T55AD   001352
T56AD   001356
T57AD   001362
T57IAD  001364
T6AD    001116
T6IAD   001120
T60AD   001366
T60IAD  001370
T61AD   001372
T61IAD  001374
T62AD   001376
T62IAD  001400
T63AD   001402
T63IAD  001404
T64AD   001406
T64IAD  001410
T7AD    001122
T7IAD   001124
UDES    000776
VECT    000610
WAM     000750
WAM0    015462
WAM01   015470
WAM1    015526
WAM2    015540
WAM2A   015574
WAM3    015604
WAM3A   015664
WAM3B   015702
WBUFF   035472
WC      000512
WCDP0   001054
WCDP2   001042
WCS1    001020
WCS2    001022
WDATA   034144
WDS     001024
WER     001026
WMSG16  034740
WMSG17  034752
WMSG2   034544
WMSG20  034757
WMSG21  034764
WMSG23  034772
WMSG24  035006
WMSG25  035024
WMSG26  035041
WMSG27  035060
WMSG28  035073
WMSG29  035111
WMSG3   034562
WMSG31  035130
WMSG32  035165
WMSG4   034603
WMSG6   034630
WMSG6A  034636
WMSG6B  034643
WMSG6C  034650
WMSG6D  034655
WMSG6E  034663
WMSG6F  034670
WMSG6G  034675
WMSG6H  034702
WMSG6I  034707
WMSG6J  034714
WMSG6K  034721
WMSG6L  034726
WMSG6M  034733
WPGFL   001000
WSICK   017126
WSIG    017274
WSIGO   017342
WSIX    017414
WTAD    000756
W1DCHK  021174
W1D0    021224
W1D0A   021242
W1D1    021260
W1D2    021326
W1D2A   021400
W1D3    021470
W1D4    021502
W1D4A   021512
W1D5    021516
W2FLG   000764
$CNTG   035175
$MNEW   035210
$MSWR   035201
$QUEST  035220
$READ   024552
.       = 036106
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*DSKM:DZTUDC,DZTUD/SOL_DSKM:DZTUCE,P11
RUN-TIME: 11 25 2 SECONDS
RUN-TIME RATIO: 184/39=4.6
CORE USED: 7K (13 PAGES)