

# TM02/TU45

CONTROL LOGIC TEST  
CZTUKA0

AH-E476A-MC

COPYRIGHT © 75-78

FICHE 1 OF 1

JUL 1978

**digital**

MADE IN USA

This microfiche card contains a grid of frames, each representing a frame of data from a control logic test. The frames are arranged in approximately 12 rows and 12 columns. Each frame contains a small, high-contrast image of a test result or data point, which is too small to read clearly. The overall appearance is that of a standard microfiche card used for data storage and retrieval.



.NLIST SEQ,LOC,BIN  
.REM\_

IDENTIFICATION

PRODUCT CODE: AC-E475A-MC  
PRODUCT NAME: CZTUKA0 TM02/TU45 CONTROL LOGIC TEST  
DATE CREATED: 25 MAY 1978  
MAINTAINER: COMPUTER SPECIAL SYSTEMS  
AUTHOR: R. B. BARNES/R. J. COLLINS

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

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

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

COPYRIGHT (a) 1975, 1976, 1977, 1978 BY DIGITAL EQUIPMENT CORPORATION



TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	1
2.	REQUIREMENTS	1
3.	LOADING PROCEEDURE	1
4.	STARTING PROCEEDURE	1
5.	SWITCH SETTINGS	2
6.	ERROR PRINTOUTS	3
7.	OPERATION	7
8.	TEST DESCRIPTION	8
9.	LISTING	



(PAGE 1)

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
  - B. 8K OF CORE
  - C. CONSOLE TTY
  - D. TM02 MAGTAPE CONTROLLER
  - E. MASSBUS CONTROLLER (RH)
  - F. TU45 MAGTAPE TRANSPORT

3. LOADING PROCEDURE

-----

USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE.

4. STARTING PROCEDURE

-----

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

- A. 200(8): STARTING AT THIS ADDRESS WILL CAUSE A PROGRAM IDENTIFICATION HEADER TO BE PRINTED BEFORE TESTING IS BEGUN.
- B. 210(8): STARTING AT THIS ADDRESS WILL NOT PRINT THE IDENTIFICATION HEADER AND IS THEREFORE GENERALLY TO BE USED FOR RESTARTS RATHER THAN INITIAL START



(PAGE 2)

5. CONSOLE SWITCH SETTINGS

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, ASSUMING A CONSOLE SWITCH REGISTER IS PRESENT. IF A CONSOLE SWITCH REGISTER IS NOT PRESENT, THE FOLLOWING PROCEDURE MUST BE IMPLEMENTED:

- A) LOAD ADDRESS 570(8) LABELLED "SWR"
- B) DEPOSIT THE VALUE 176(8)
- C) LOAD ADDRESS 176(8)
- D) DEPOSIT THE DESIRED SWITCH VALUE.

- SW15: 1=HALT ON ERROR  
0=CONTINUE
- SW14: 1=LOOP ON ERROR (SCOPE)  
0=CONTINUE
- SW13: 1=DO NOT PRINT ERRORS  
0=PRINT ALL ERRORS
- SW12: 1=INHIBIT ITERATIONS  
0=ITERATE EACH TEST ITS ASSIGNED AMOUNT
- SW11: 1=DO CONTINUOUS CYCLE  
0=HALT AT END OF PASS
- SW10: 1=HALT AT END OF CURRENT TEST  
0=CONTINUE TO NEXT TEST
- SW9: 1=DO MANUAL INTERVENTION TESTS  
0=INHIBIT MANUAL INTERVENTION
- SW8: 1=INHIBIT WRAP AROUND DATA CHECK  
0=DO DATA CHECKS
- SW7: 1=INHIBIT WRAP AROUND STATUS CHECK  
0=DO STATUS CHECK
- SW6: 1=SELECTABLE WRAP DATA PATTERN (IN SINGLE TEST)  
0=AUTO PATTERN
- SW5-0: SELECT INDIVIDUAL TEST (1-64)\*\* 00=DO ALL TESTS



(PAGE 3)

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).



(PAGE 4)

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

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

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

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

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

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

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

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



(PAGE 5)

LOGIC TEST 30: DTE (M8906 RH)

UNEXPECTED ERROR BITS

CS1	WC	BA	FC	CS2	DS	ER	AS	MR	TC
144260	002006	006600	000000	001300	150600	030000	000001	000017	100300

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

LOGIC TEST 32: UNS (M8909)

NOT RESET BY DRIVE CLEAR

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

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



(PAGE 6)

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)



(PAGE 7)

7. OPERATION  
-----

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

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

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

SINGLE TEST SELECTION: (SW0-SW5)

WHEN SW0-SW5 ARE SET TO ZERO (00), THE SCHEDULAR WILL EXECUTE ALL TESTS (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.

(PAGE 8)

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: M5904,CABLE,M5903,M8909

CIRCUITS

PRINT REFERENCES

RH-DS BITS	(CSRB)
RH-NED BIT	(CSRB)
MASSBUS CABLE C(DEM,TRA,DS BITS)	(MB3)
DRIVE ADDRESS	(MBI2)
DEM-TRA HANDSHAKE	

LOGIC TEST #2: REGISTER ADDRESSING

PURPOSE: CHECK THE REGISTER SELECT LINES

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

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8909,M8905YA,M8903YA

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)



(PAGE 9)

LOGIC TEST #3: CONTROL BUS  
-----

PURPOSE: VERIFY THAT ALL CONTROL LINES PROPERLY TRANSMIT ONES AND ZEROS.

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

LIKELY FAULT LOCATIONS: M5904,CABLES,M5903YA,M8909,M8905YA,M8903YA  
-----

CIRCUITS  
-----

PRINT REFERENCE  
-----

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

LOGIC TEST #4: SLAVE ADDRESSING  
-----

PURPOSE: VERIFY THE FUNCTIONING OF THE SLAVE ADDRESS BITS IN THE TAPE CONTROL REGISTER THE SLAVE ADDRESS BUS LINES, THE ADDRESS DECODE CIRCUIT IN THE TU45 AND THE SPR BIT.

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

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

LIKELY FAULTS LOCATIONS: M8905YA,M8907,CABLE,M8921,M8903YA  
-----

CIRCUITS  
-----

PRINT REFERENCE  
-----

REGISTER SELECT	(MBI2)
SLAVE ADDRESS BITS	(MR6)
SLAVE ADDRESS LINES	(M8907,2-2),(LAW6)
TU45 ADDRESS DECODE	(M8921)
SPR BIT	(M8921)(TCCM7)

(PAGE 10)

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: M8905YA  
-----

CIRCUITS  
-----

PRINT REFERENCE  
-----

C-LINES  
MAINTENANCE REGISTER  
M.R. FUNCTION DECODE  
M.R. MULTIPLEXOR

(MR2,3,5)  
(MR5)  
(MR4)

LOGIC TEST #6: TAPE CONTROL REGISTER BITS  
-----

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

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

LIKELY FAULT LOCATIONS: M8909,M8905YA  
-----

CIRCUITS  
-----

PRINT REFERENCE  
-----

TM02 REGISTER SELECT  
TC FLIP-FLOPS, MULTIPLEXERS

(MB12)  
(MR6)



(PAGE 11)

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, M8905YA  
-----

CIRCUITS  
-----

PRINT REFERENCE  
-----

TM02 REGISTER SELECTION	(MBI2)
FUNCTION CODE FLOPS	(MBI5)
FUNCTION CODE MULTIPLEXERS	(MR6)

(PAGE 12)

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,M8905YA  
-----

CIRCUIT  
-----

PRINT REFERENCE  
-----

FCS	MB18
SET ILF	MB17
SET NEF	MB17
GO BIT	MB15
GO BIT MULTIPLEXER	MR6
SET ILR	MB12

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  
-----



(PAGE 13)

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: M8921,SLAVE CABLE, M8903YA  
-----

CIRCUIT  
-----

PRINT REFERENCE  
-----

MOL  
WRL  
DPR  
DRY  
BOT

TCCM7,M8908,M8921  
TCCM7,M8908,M8921  
TCCM7  
TCCM7  
TCCM7,M8908YA,M8921

(PAGE 14)

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: M8921,M8903YA,M8909,SLAVE CABLE  
-----

CIRCUIT  
-----

PRINT REFERENCE  
-----

SSC  
ATA

M8921,TCCM7  
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: M8921,SLAVE CABLE,M8903YA  
-----

CIRCUIT  
-----

PRINT REFERENCE  
-----

SSC  
EOT  
SLA

M8921,TCCM7  
TCCM7,M8908YA,M8921  
TCCM7,M8921,YC,M8908



(PAGE 15)

LOGIC TEST #17: STATUS AT ONLINE LOADED

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

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

LOGIC TEST #20: ILLEGAL FUNCTION

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

LIKELY FAULT LOCATION: M8909

CIRCUIT

PRINT REFERENCE

SET ILF DECODE  
ILF FLOP  
ILF MULTIPLEXER

MBI5,MBI7  
MBI11  
MBI10

(PAGE 16)

LOGIC TEST #21: REGISTER MODIFICATION REFUSED  
-----

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

LIKELY FAULT LOCATION: M8909  
-----

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

LOGIC TEST #22: CONTROL BUS PARITY (CPAR)  
-----

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

LIKELY FAULT LOCATIONS: M8909  
-----

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

(PAGE 17)

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: M8905YA, 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, M8905YA, M8906  
-----

CIRCUIT -----	PRINT REFERENCE -----
MM CLK	MR5
WRT CLK GENERATION	TCCM4
DPAR FLOP	MBI11
DATA BUS PARITY TREE	BF3



(PAGE 18)

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

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

LIKELY FAULT LOCATION: M8909  
-----

CIRCUIT  
-----

PRINT REFERENCE  
-----

NEF FLOP  
NEF MULTIPLEXER  
SET NEF

MB111  
MB110  
MB17

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, M8903YA, M8905YA  
-----

CIRCUITS  
-----

PRINT REFERENCE  
-----

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

MB1  
MB19  
MB111  
TCCM5  
MR5

(PAGE 19)

LOGIC TEST #27: ILLEGAL REGISTER  
-----

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

LIKELY FAULT LOCATIONS: MASSBUSS, M8909  
-----

CIRCUITS PRINT REFERENCE  
-----

REGISTER SELECT LINES	MB1, MB2
REGISTER SELECT DECODE	MB12
ILR FLOP	MB111

LOGIC TEST #30: DRIVE TIMING ERROR  
-----

PROGRAMMED SEQUENCE:  
-----

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

LIKELY FAULT LOCATIONS: M8909, M8905YA, M8906, MB CABLES  
-----

CIRCUITS PRINT REFERENCES  
-----

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

(PAGE 20)

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: M8903YA, M8909

CIRCUITS

PRINT REFERENCES

OPI TIMER  
OPI FLOP  
OPI TIMER CONTROL

TCCM5  
MBI11  
MBI7

LOGIC TEST 32: UNSAFE (UNS)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8909, M8921, SLAVE CABLE

CIRCUITS

PRINT REFERENCES

UNSAFE FLOP  
SET UNSAFE  
MOL GENERATION

MBI11  
MBI7  
M8921



(PAGE 21)

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, M8903YA

CIRCUITS

PRINT REFERENCES

SPACE FUNCTION DECODE  
PIP GENERATION  
STATUS REGISTER

MB15  
TCCM7  
TCCM7

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905YA, SLAVE BUS, M8921, M8903YA

CIRCUITS

PRINT REFERENCES

DENSITY BITS  
DENSITY LINES  
PES CIRCUIT  
PES STATUS BIT

MR6  
SBC  
SC3  
TCCM7

(PAGE 22)

LOGIC TEST 35: TAPE CONTROL WRITE (TCW)  
-----

PROGRAMMED SEQUENCE:  
-----

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

LIKELY FAULT LOCATION: M8905YA  
-----

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, M8903YA  
-----

CIRCUITS  
-----

PRINT REFERENCES  
-----

FCS BIT  
FCS MULTIPLEXER

MB18  
TCCM7

(PAGE 23)

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: M8903YA, M8921  
-----

CIRCUITS  
-----

PRINT REFERENCES  
-----

ACCL BIT, MOTION DELAY COUNTER CLOCK	TCCM3 SC2
---	--------------

LOGIC TEST 40: PE TAPE MARK (TM)  
-----

PROGRAMMED SEQUENCE:  
-----

DRIVE IS INITIALIZED, WAMO IS SET, WRITE TAPE MARK IS SET. AFTER DELAY TAPE MARK BIT IS CHECKED. WAMO MULTIPLEXES THE OUTPUT OF THE WRITE DATA GENERATOR ONTO THE RDA LINES. THE DATA SYNC MODULES SYNC ON THE DATA AND SEND ENVELOPE INFORMATION TO THE TAPE MARK DETECTOR ON M8902YA.

LIKELY FAULT LOCATIONS: M8902YA, M8901YA, M8903YA, M8905YA  
-----

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	MB15
WAMO SIGNAL	MR5



(PAGE 24)

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

PROGRAMMED SEQUENCE:  
-----

SAME AS TEST 40 EXCEPT NRZ DENSITY IS SELECTED.

LIKELY FAULT LOCATIONS: M8903YA, M8904  
-----

CIRCUITS -----	PRINT REFERENCES -----
WRITE DATA BUFFER	TCCM2
RSDO 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, M8905YA, MASSBUS P-LINES  
-----

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

(PAGE 25)

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 MAINTENANCE REGISTER WITH EOR CLR. THE SEQUENCE IS REPEATED WITH VARYING DATA PATTERNS.

LIKELY FAULT LOCATIONS: M8906, M8905YA, M8903YA  
-----

CIRCUITS  
-----

PRINT REFERENCES  
-----

BIT FIDDLER CHAR UNPACK  
BIT FIDDLER DATA REQUEST  
WRT STRB.  
MAINT. REG. DATA FIELD

BF4  
BF2  
TCCM4  
MR2, MR3

(PAGE 26)

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 M8902YA 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: M8903YA, M8904 (CRC GENERATOR)  
-----

CIRCUITS  
-----

PRINT REFERENCES  
-----

WRITE BUFFER  
CRC GENERATOR

TCCM2  
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: M8902YA (WRITE CONTROL STATES), M8903YA  
-----

CIRCUITS  
-----

PRINT REFERENCE  
-----

WRITE BUFFER  
WRITE CONTROL

TCCM2  
TCPE3



(PAGE 27)

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, M8903YA  
-----

CIRCUITS  
-----

PRINT REFERENCES  
-----

CRC GENERATOR	CNR22
CRC CHECKOUT	CNR23
CRC, CRC STROBE	TCCM4
READ LINE MULTIPLEXERS	TCCM6
MM CLK	MR5
CRC READ TIMING	CNR24
SHUTDOWN CIRCUITRY	TCCM5

LOGIC TEST 51: WRP0, PE, NORMAL, ODD  
-----

REPEAT OF TEST 50 IN PE MODE.

LIKELY FAULT LOCATIONS: M8901YA, M8902YA, M8903YA  
-----

CIRCUITS  
-----

PRINT REFERENCES  
-----

DATA DISCRIMINATOR	DS2, DS4, DS6
PHASE LOCKED CLOCK	DS3, DS5, DS7
SKEW BUFFER	DS3, DS5, DS7
PE WRITE MAJOR STATES	TCPE3
PE READ MAJOR STATES	TCPE5
WRAP 0 CIRCUIT TO BLOCK RLT RDS	TCPE3
DESKEW BUFFER READ COUNTER	TCPE4

(PAGE 28)

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: M8903YA

LOGIC TEST 55: EVEN PARITY READ: WAM0,

REPEAT OF TEST 50 EXCEPT EVEN PARITY IS USED.

LIKELY FAULT LOCATIONS: M8903YA, M8904

(PAGE 29)

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 TMO2 LOGIC INDEPENDENT OF THE TU45. HOWEVER THERE ARE A FEW SIGNALS WHICH ARE REQUIRED FROM THE TU45 TO COMPLETE THE TESTS, AND AT LEAST ONE CASE WHERE TU45 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 WAMO TO GENERATE DATA AND REC(SB)L
4. RSD0(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 READ CHAIN.



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

LIKELY FAULT LOCATIONS: M8905YA, M8904, M8921, SLAVE CABLE,  
M8910

CIRCUITS

MM FUNCTION DECODE  
CRC CHECK CIRCUIT

PRINT REFERENCES

MRS  
CNRZ3

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 CORRECTABLE DATA ERROR. THE DEAD TRACK REGISTER IS CHECKED FOR BIT 1.

LIKELY FAULT LOCATIONS: M8905YA, M8901YA, M8902YA  
-----

CIRCUITS  
-----

PRINT REFERENCES  
-----

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

LOGIC TEST 62: PE INCORRECTABLE DATA  
-----

REPEAT OF TEST 61, EXCEPT THAT THE MAINT. MODE FUNCTION GOUNDS 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: M8902YA, M8901YA  
-----

CIRCUIT  
-----

PRINT REFERENCE  
-----

INC ERROR, PEF,	TCPE2
-----------------	-------

(PAGE 32)

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: M8902YA, M8903YA, M8905YA  
-----

CIRCUITS  
-----

PRINT REFERENCES  
-----

PEF.  
WRITE BUFFER  
MM DECODE

TCPE2  
TCCM2  
MRS

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



(PAGE 33)

9. LISTING

-

```
1372          .LIST  SEQ,LOC,BIN
1373          .TITLE TM02/TU45 CONTROL LOGIC TEST
1374          :CZTUKAO
1375          :25 MAY 78
1376          :R. BARNES/R. J. COLLINS
1377
1378          .ENABLE ABS
1379
1380
1381          :CONSOLE SWITCHES*****
1382          :
1383          :SW15: 1=HALT ON ERROR
1384          :      0=CONTINUE
1385          :SW14: 1=LOOP ON ERROR
1386          :      0=CONTINUE
1387          :SW13: 1=DO NOT PRINT ERRORS
1388          :      0=PRINT ERRORS
1389          :SW12: 1=INHIBIT ITERATIONS
1390          :      0=DO ITERATIONS
1391          :SW11: 1=CONTINUOUS CYCLE
1392          :      0=HALT AT END OF PASS
1393          :SW10: 1=HALT AT END OF EACH TEST
1394          :      0=CONTINUE
1395          :SW9:  11=DO MANUAL INTERVENTION TESTS
1396          :      0=INHIBIT MANUAL INTERVENTION
1397          :SW8:  1=NO WRAP DATA CHECK
1398          :      0=DO WRAP DATA CHECK
1399          :SW7:  1=NO WRAP STATUS CHECK
1400          :      0=DO WRAP STATUS CHECK
1401          :SW6:  1=SELECTABLE WRAP DATA PATTERN (IN SINGLE TEST)
1402          :      0=AUTO PATTERNS
1403          :SW0-5: SELECT TEST NUMBER :: 00=ALL TESTS
```





```
1450                                     ;REGISTER EQUIVS*****
1451
1452          000000                      R0=%0
1453          000001                      R1=%1
1454          000002                      R2=%2
1455          000003                      R3=%3
1456          000004                      R4=%4
1457          000005                      R5=%5
1458          000006                      SP=%6
1459          000007                      PC=%7
1460
1461                                     ;TRAP CATCHERS*****
1462
1463          000000                      .=0
1464          000200                      .REPT 200
1465
1466          .+2
1467          HALT
1468          .ENDR
1469
1470                                     ;TTY INTERRUPT VECTOR*****
1471
1472          000060          000060          .=60
1473          000062          024316          TTINT          ;TTY INTERRUPT HEADER ADDRESS
1474          0
1475
1476                                     ;START ADDRESS*****
1477
1478          000200          000200          .=200
1479          000202          000167          001206          CLR          R0
1480
1481                                     ;PROGRAM START
1482          000210          000240          JMP          START
1483          000212          012700          000001          . =210
1484          000216          000167          001172          NOP
1485
1486                                     ;SET NO HEADER FLAG
1487          .
1488                                     ;TM02 INTERRUPT VECTOR*****
1489          000224          000224          .=224
1490          000226          024302          MTINT          ;TAPE INTERRUPT HANDLER ADDRESS
1491          000340          000340          340
```

```

1492
1493          000510
1494          .=-510
1495          ;MASS BUS REGISTER EQUIVS*****
1496 000510 172440      C1: 172440
1497 000512 172442      WC: 172442
1498 000514 172444      BA: 172444
1499 000516 172446      FC: 172446
1500 000520 172450      CS: 172450
1501 000522 172452      DS: 172452
1502 000524 172454      ER: 172454
1503 000526 172456      AS: 172456
1504 000530 172460      CC: 172460
1505 000532 172462      DB: 172462
1506 000534 172464      MR: 172464
1507 000536 172466      DT: 172466
1508 000540 172470      SN: 172470
1509 000542 172472      TC: 172472
1510
1511          ;ILLEGAL FUNCTION CODES
1512
1513 000544 005405      ILFT: 5405
1514 000546 007415      7415
1515 000550 016423      16423
1516 000552 020437      20437
1517 000554 022443      22443
1518 000556 025447      25447
1519 000560 031455      31455
1520 000562 033465      33465
1521 000564 036473      36473
1522
1523          ;CONSTANTS*****
1524
1525 000566 177776      PSW: 177776      ;PROCESSOR STATUS
1526 000570 177570      SWR: 177570      ;SWITCH REGISTER
1527 000572 177560      TKS: 177560      ;TTY READER STATUS
1528 000574 177562      TKB: 177562      ;TTY READ BUFFER
1529 000576 177564      TPS: 177564      ;TTY PUNCH STATUS
1530 000600 177566      TPB: 177566      ;TTY PUNCH BUFFER
1531 000602 177777      SERNUM: 177777   ;SERIAL NUMBER
1532 000604 000011      DRVTP: 011       ;DRIVE TYPE
1533 000606 000020      ITAMT: 20        ;ITERATION AMOUNT
1534 000610 000224      VECT: 224        ;INTERRUPT VECTOR(RH)
1535 000612 172440      REGS: 172440     ;STARTING REGISTER ADDRESS
  
```

```
1536                                     :FLAGS AND COUNTERS*****
1537
1538 000614 000000 TOB: 0
1539 000616 000000 TIB: 0
1540 000620 000000 HDRFL: 0
1541 000622 000000 EMADDR: 0
1542 000624 000000 DRVN: 0
1543 000626 000000 TRO0: 0
1544 000630 000000 TRO1: 0
1545 000632 000000 TRO2: 0
1546 000634 000000 TRO3: 0
1547 000636 000000 TRO4: 0
1548 000640 000000 TRO5: 0
1549 000642 000000 TRO6: 0
1550 000644 000000 TRO7: 0
1551 000646 000000 TR10: 0
1552 000650 000000 TR11: 0
1553 000652 000000 TR12: 0
1554 000654 000000 TR13: 0
1555 000656 000000 TR14: 0
1556 000660 000000 TR15: 0
1557 000662 000000 NRZOF: 0
1558 000664 000000 SLVN: 0
1559 000666 000000 PFLG: 0
1560 000670 000000 RTRN: 0
1561 000672 000000 ERADD: 0
1562 000674 000000 TEMP1: 0
1563 000676 000000 TEMP2: 0
1564 000700 000000 TEMP3: 0
1565 000702 000000 ITCNT: 0
1566 000704 000000 SAV1: 0
1567 000706 000000 SAV2: 0
1568 000710 000000 SAV3: 0
1569 000712 000000 SCOLP: 0
1570 000714 000000 ITRLP: 0
1571 000716 000000 EXFL: 0
1572 000720 000000 ATAF: 0
1573 000722 000000 SLAF: 0
1574 000724 000000 SSCF: 0
1575 000726 000000 ERRF: 0
1576 000730 000000 ASF: 0
1577 000732 000000 SCF: 0
1578 000734 000000 TREF: 0
1579 000736 000000 PEXFL: 0
1580 000740 000000 STFLG: 0
1581 000742 000000 LTADD: 0
1582 000744 000000 T24FL: 0
1583 000746 000000 ADDFL: 0
1584 000750 000000 WAM: 0
1585 000752 000000 FUN: 0
1586 000754 000000 DATC: 0
1587 000756 000000 WTAD: 0
1588 000760 000000 DATAD: 0
1589 000762 000000 RDAD: 0
1590 000764 000000 W2FLG: 0
1591 000766 000000 DERFL: 0
```



1592	000770	000000	PREFL:	0	
1593	000772	000000	SERFL:	0	
1594	000774	000000	CRCNT:	0	
1595	000776	000000	UDES:	0	
1596	001000	000000	WPGFL:	0	
1597	001002	000000	PATRN:	0	
1598	001004	000000	STATF:	0	
1599	001006	000000	RDRVF:	0	
1600	001010	000000	RCDP:	0	
1601	001012	000000	STATC:	0	
1602	001014	000000	SKAT:	0	
1603	001016	000000	PCNTR:	C	;PASS COUNTER
1604					
1605					;EXPT WRAP STATUS*****
1606					
1607	001020	000000	WCS1:	0	
1608	001022	000000	WCS2:	0	
1609	001024	000000	WDS:	0	
1610	001026	000000	WER:	0	
1611					
1612					;DATA PATTERN GENERATORS*****
1613					
1614	001030	000000	DATBL:	0	
1615	001032	016026	DATA0:	DAT1	;ALL ONE BITS
1616	001034	016050	DATA1:	DAT2	;ALL ZERO BITS
1617	001036	016056	DATA2:	DAT3	;ALTERNATING ONE/ZERO BITS
1618	001040	016066	DATA3:	DAT4	;ALL BITS 0-377
1619					
1620					;CORE DUMP PATTERNS*****
1621					
1622	001042	000005	WCDP2:	5	
1623	001044	000005		5	
1624	001046	000012		12	
1625	001050	000012		12	
1626	001052	000000		0	
1627	001054	000017	WCDP0:	17	
1628	001056	000017		17	
1629	001060	000017		17	
1630	001062	000017		17	
1631	001064	000000		0	

1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687

001066 000000  
001070 000000  
001072 002214  
001074 002214  
001076 002444  
001100 002444  
001102 002674  
001104 002676  
001106 003060  
001110 003060  
001112 003346  
001114 003356  
001116 003626  
001120 003630  
001122 004054  
001124 004056  
001126 004206  
001130 004210  
001132 004350  
001134 004352  
001136 004626  
001140 004630  
001142 005046  
001144 005060  
001146 005160  
001150 005222  
001152 005316  
001154 005360  
001156 005440  
001160 005502  
001162 005562  
001164 005624  
001166 005716  
001170 005734  
001172 006070  
001174 006106  
001176 006240  
001200 006256  
001202 006372  
001204 006410  
001206 006526  
001210 006542  
001212 007052  
001214 007060  
001216 007204  
001220 007212  
001222 007420  
001224 007444  
001226 007536  
001230 007560  
001232 010054  
001234 010070  
001236 010404

:LOGIC TEST ENTRY TABLE\*\*\*\*\*

TSTTBL: 0  
0  
T1AD: LT1  
T1IAD: LT1  
T2AD: LT2  
T2IAD: LT2  
T3AD: LT3  
T3IAD: LT3IT  
T4AD: LT4  
T4IAD: LT4  
T5AD: LT5  
T5IAD: LT5IT  
T6AD: LT6  
T6IAD: LT6IT  
T7AD: LT7  
T7IAD: LT7IT  
T10AD: LT10  
T10IAD: LT10IT  
T11AD: LT11  
T11IAD: LT11IT  
T12AD: LT12  
T12IAD: LT12IT  
T13AD: LT13  
T13IAD: LT13IT  
T14AD: LT14  
T14IAD: LT14IT  
T15AD: LT15  
T15IAD: LT15IT  
T16AD: LT16  
T16IAD: LT16IT  
T17AD: LT17  
T17IAD: LT17IT  
T20AD: LT20  
T20IAD: LT20IT  
T21AD: LT21  
T21IAD: LT21IT  
T22AD: LT22  
T22IAD: LT22IT  
T23AD: LT23  
T23IAD: LT23IT  
T24AD: LT24  
T24IAD: LT24IT  
T25AD: LT25  
T25IAD: LT25IT  
T26AD: LT26  
T26IAD: LT26IT  
T27AD: LT27  
T27IAD: LT27IT  
T30AD: LT30  
T30IAD: LT30IT  
T31AD: LT31  
T31IAD: LT31IT  
T32AD: LT32

1688	001240	010420	T32IAD:	LT32IT
1689	001242	010536	T33AD:	LT33
1690	001244	010552	T33IAD:	LT33IT
1691	001246	010640	T34AD:	LT34
1692	001250	010662	T34IAD:	LT34IT
1693	001252	011002	T35AD:	LT35
1694	001254	011016	T35IAD:	LT35IT
1695	001256	011154	T36AD:	LT36
1696	001260	011170	T36IAD:	LT36IT
1697	001262	011274	T37AD:	LT37
1698	001264	011310	T37IAD:	LT37IT
1699	001266	011444	T40AD:	LT40
1700	001270	011466	T40IAD:	LT40IT
1701	001272	011572	T41AD:	LT41
1702	001274	011606	T41IAD:	LT41IT
1703	001276	012040	T42AD:	LT42
1704	001300	012040		LT42
1705	001302	012162	T43AD:	LT43
1706	001304	012162		LT43
1707	001306	012252	T44AD:	LT44
1708	001310	012252		LT44
1709	001312	012370	T45AD:	LT45
1710	001314	012370		LT45
1711	001316	012460	T46AD:	LT46
1712	001320	012460		LT46
1713	001322	012600	T47AD:	LT47
1714	001324	012600		LT47
1715	001326	012674	T50AD:	LT50
1716	001330	012674		LT50
1717	001332	013012	T51AD:	LT51
1718	001334	013012		LT51
1719	001336	013102	T52AD:	LT52
1720	001340	013102		LT52
1721	001342	013224	T53AD:	LT53
1722	001344	013224		LT53
1723	001346	013360	T54AD:	LT54
1724	001350	013360		LT54
1725	001352	013436	T55AD:	LT55
1726	001354	013436		LT55
1727	001356	013514	T56AD:	LT56
1728	001360	013514		LT56
1729	001362	013600	T57AD:	LT57
1730	001364	013632	T57IAD:	LT57IT
1731	001366	002032		TSCD2
1732	001370	002032		TSCD2
1733	001372	014402	T61AD:	LT61
1734	001374	014432	T61IAD:	LT61IT
1735	001376	014666	T62AD:	LT62
1736	001400	014714	T62IAD:	LT62IT
1737	001402	015140	T63AD:	LT63
1738	001404	015170	T63IAD:	LT63IT
1739	001406	015420	T64AD:	LT64
1740	001410	015436	T64IAD:	LT64IT
1741	001412	000000	TADX:	0



```

1742
1743
1744
1745 001414 012777 000340 177144 START: MOV #340,@PSW ;SET PRIORITY
1746 001422 012706 000500 MOV #500,SP ;SET STACK POINTER
1747 001426 005700 TST R0
1748 001430 001136 BNE ST2
1749 001432 005067 177356 CLR SKAT ;CLEAR SKIP ADDRESS TEST FLAG
1750 001436 012704 025350 MOV #MSG1,R4
1751 001442 004767 023116 JSR PC,TTOUT ;PRINT TITLE
1752 001446 012704 027210 MOV #MSG44,R4
1753 001452 004767 023106 JSR PC,TTOUT ;REQUEST REGISTER ADDRESS
1754 001456 016703 177130 MOV REGS,R3
1755 001462 004767 023224 JSR PC,OCTP ;PRINT CURRENT ADDRESS
1756 001466 012705 000612 MOV #REGS,R5 ;SET ADDRESS SAVE LOC
1757 001472 012701 000006 MOV #6,R1 ;SET SIZE OF RESPONSE
1758 001476 012702 177000 MOV #177000,R2 ;SET UPPER LIMIT
1759 001502 012703 160000 MOV #160000,R3 ;SET LOWER LIMIT
1760 001506 004767 022614 JSR PC,TTR ;GO GET RESPONSE
1761 001512 012704 027232 MOV #MSG45,R4
1762 001516 004767 023042 JSR PC,TTOUT ;REQUEST VECTOR
1763 001522 016703 177062 MOV VECT,R3
1764 001526 004767 023160 JSR PC,OCTP ;PRINT CURRENT VECTOR
1765 001532 012705 000610 MOV #VECT,R5 ;SET ADDRESS SAVE LOC
1766 001536 012701 000003 MOV #3,R1 ;SET SIZE OF RESPONSE
1767 001542 012702 000224 MOV #224,R2 ;SET UPPER LIMIT
1768 001546 012703 000150 MOV #150,R3 ;SET LOWER LIMIT
1769 001552 004767 022550 JSR PC,TTR ;GO GET RESPONSE
1770 001556 016700 177026 MOV VECT,R0 ;GET VECTOR
1771 001562 012720 024302 MOV #MTINT,(R0)+ ;LOAD INTERRUPT ADDRESS IN VECTOR
1772 001566 012710 000340 MOV #340,(R0) ;LOAD PRIORITY
1773 001572 016700 177014 MOV REGS,R0 ;GET START OF REGS
1774 001576 012701 000016 MOV #16,R1 ;SET NUMBER OF REGS
1775 001602 012702 000510 MOV #C1,R2 ;GET START OF TABLE
1776 001606 010022 ST0: MOV R0,(R2)+ ;BUILD TABLE
1777 001610 062700 000002 ADD #2,R0 ;BUMP ADDRESS
1778 001614 005301 DEC R1 ;SEE IF DONE
1779 001616 001373 BNE ST0 ;IF NOT: BR
1780 001620 012702 000614 MOV #TOB,R2
1781 001624 012700 000077 MOV #77,R0
1782 001630 005022 ST1: CLR (R2)+ ;CLEAR FLAGS + COUNTERS
1783 001632 005300 DEC R0
1784 001634 001375 BNE ST1
1785 001636 012704 027460 MOV #MSG52,R4
1786 001642 004767 022716 JSR PC,TTOUT ;PRINT NRZ ONLY REQUEST
1787 001646 012705 000662 MOV #NRZOF,R5
1788 001652 012701 000001 MOV #1,R1 ;SET SIZE OF ENTRY
1789 001656 012702 000001 MOV #1,R2 ;SET UPPER LIMIT
1790 001662 012703 000000 MOV #0,R3 ;SET LOWER LIMIT
1791 001666 004767 022434 JSR PC,TTR ;GO GET RESPONSE
1792 001672 012704 027567 MOV #MSG56,R4
1793 001676 004767 022662 JSR PC,TTOUT ;REQUEST STATIC ONLY
1794 001702 012705 001012 MOV #STATC,R5 ;SET ADDRESS OF STATIC FLAG
1795 001706 012701 000001 MOV #1,R1 ;SET SIZE OF RESPONSE
1796 001712 012702 000001 MOV #1,R2 ;SET UPPER LIMIT
1797 001716 012703 000000 MOV #0,R3 ;SET LOWER LIMIT

```

1798 001722 004767 022400  
1799 001726 005067 177064

ST2: JSR PC,TTR ;GET RESPONSE  
CLR PCNTR ;CLEAR PASS COUNTER



```

1800
1801                                     ;TEST SCHEDULAR*****
1802
1803 001732 005067 177042          TSCD: CLR      WPGFL          ;CLEAR WRAP PATRN FLAG
1804 001736 005067 176776          CLR      STFLG          ;CLEAR SINGLE TEST FLAG
1805 001742 017700 176622          MOV      @SWR,R0
1806 001746 042700 177700          BIC      #177700,R0
1807 001752 005700
1808 001754 001046
1809 001756 012767 001066 176756   MOV      #TSTTBL,LTADD ;GO SELECT SINGLE TEST
1810 001764 062767 000004 176750   TSCD0: ADD      #4,LTADD
1811 001772 016767 176744 176714   MOV      LTADD,IIRLP
1812 002000 062767 000002 176706   ADD      #2,IIRLP      ;SET ITERATION ADDRESS
1813 002006 005777 176730          TST      @LTADD
1814 002012 001002
1815 002014 000167 000124          BNE      TSCD1
1816 002020 005067 176574          JMP      TEND          ;GO TO END ROUTINE
1817 002024 017700 176712          TSCD1: CLR      HDRFL          ;CLEAR PRINT HEADER FLAG
1818 002030 000110          MOV      @LTADD,R0     ;SET POINTER TO TEST
1819 002032 032777 002000 176530   TSCD2: JMP      (R0)        ;GO TO TEST
1820 002040 001403          BIT      #2000,@SWR    ;SEE IF HALT ON TEST
1821 002042 000000          BEQ      TSCD3        ;IF NOT: BR
1822 002044 005067 176730          HALT
1823 002050 005767 176664          TSCD3: CLR      WPGFL          ;CLEAR WRAP DATA GENERATOR FLAG
1824 002054 001743          TST      STFLG        ;SE IF SINGLE TEST
1825 002056 017700 176506          BEQ      TSCD0        ;IF NOT: BR
1826 002062 042700 177700          MOV      @SWR,R0
1827 002066 005700          BIC      #177700,R0   ;MASK TEST NUMBER
1828 002070 001720          TST      RO          ;SEE IF RETURN TO ALL
1829 002072 012767 000001 176640   STSCD: BEQ      TSCD      ;IF SO: BR
1830 002100 022700 000065          MOV      #1,STFLG     ;SET SINGLE TEST FLAG
1831 002104 003417          CMP      #65,R0       ;SEE IF EXCEEDED TESTS
1832 002106 000241          BLE      TEND        ;IF SO: BR
1833 002110 006100          CLC
1834 002112 006100          ROL      RO
1835 002114 012767 001066 176620   MOV      #TSTTBL,LTADD ;SET TABLE MODIFIER
1836 002122 060067 176614          ADD      RO,LTADD     ;SET TEST POINTER
1837 002126 016767 176610 176560   MOV      LTADD,IIRLP
1838 002134 062767 000002 176552   ADD      #2,IIRLP     ;SET ITERATION POINTER
1839 002142 000726          BR       TSCD1
1840 002144 012704 027050          TEND: MOV      #MSG41,R4
1841 002150 004767 022410          JSR      PC,TTOUT     ;PRINT END OF PASS
1842 002154 016703 176636          MOV      PCNTR,R3
1843 002160 004767 022526          JSR      PC,OCTP      ;PRINT PASS NUMBER
1844 002164 032777 004000 176376   BIT      #4000,@SWR   ;SEE IF HALT ON PASS
1845 002172 001001          BNE      TENDX        ;IF NOT: BR
1846 002174 000000          HALT
1847 002176 012767 000001 176610   TENDX: MOV      #1,SKAT   ;SET SKIP ADDRESS TEST FLAG
1848 002204 005267 176606          INC      PCNTR        ;BUMP PASS COUNTER
1849 002210 000167 177516          JMP      TSCD         ;RESTART
    
```



```

1850 ;LOGIC TEST 1: DRIVE ADDRESSING*****
1851
1852 002214 005767 176574 LT1: TST SKAT ;SEE IF SKIP ADDRESS TESTS
1853 002220 001403 BEQ LT1G0 ;IF NOT: BR
1854 002222 005767 176512 TST STFLG ;SEE IF SINGLE TEST
1855 002226 001504 BEQ LT1X ;IF NOT: BR
1856 002230 012704 025473 LT1G0: MOV #MSG2A,R4
1857 002234 004767 022324 JSR PC,TTOUT ;PRINT TEST INSTRUCTIONS
1858 002240 012767 027614 176354 LT1G: MOV #MSLT1,EMADDR ;SET HEADER ADDRESS
1859 002246 012704 025454 MOV #MSG2,R4
1860 002252 004767 022306 JSR PC,TTOUT ;REQUEST DRIVE NUMBER
1861 002256 012705 000624 MOV #DRVN,R5
1862 002262 012701 000001 MOV #1,R1
1863 002266 012702 000007 MOV #7,R2
1864 002272 012703 000000 MOV #0,R3
1865 002276 004767 022024 JSR PC,TTR ;GET DRIVE NUMBER
1866 002302 005767 176366 TST TEMP1 ;SEE IF ANOTHER DRIVE
1867 002306 001454 BEQ LT1X ;IF NOT: BR
1868 002310 005001 CLR R1 ;SELECT DRIVE 0
1869 002312 012700 000010 MOV #10,R0 ;SET NUMBER OF DRIVES
1870 002316 012777 000040 176174 LT1A: MOV #40,ACS ;INIT
1871 002324 010177 176170 MOV R1,ACS ;SELECT DRIVE
1872 002330 005777 176154 TST @C1 ;ACCESS DRIVE
1873 002334 032777 010000 176156 BIT #10000,ACS ;SEE IF NED
1874 002342 001005 BNE LT1B ;IF SO: BR
1875 002344 026701 176254 CMP DRVN,R1 ;SEE IF SHOULD BE NED
1876 002350 001407 BEQ LT1C ;IF NOT: BR
1877 002352 000167 000022 JMP LT1ER ;ELSE GO TO ERROR
1878 002356 026701 176242 LT1B: CMP DRVN,R1 ;SEE IF SHOULD BE NED
1879 002362 001002 BNE LT1C ;IF SO: BR
1880 002364 000167 000020 JMP LT1ER1 ;ELSE GO TO ERROR
1881 002370 005300 LT1C: DEC R0
1882 002372 001722 BEQ LT1G ;IF DONE ALL: BR
1883 002374 005201 INC R1 ;SELECT NEXT DRIVE
1884 002376 000747 BR LT1A ;CONTINUE
1885 002400 012767 000001 176310 LT1ER: MOV #1,EXFL ;FLAG EXPT
1886 002406 000403 BR LT1ER2
1887 002410 012767 000002 176300 LT1ER1: MOV #2,EXFL ;FLAG NOT EXPT
1888 002416 012767 025622 176246 LT1ER2: MOV #MSG3,ERADD ;FLAG CONDITION
1889 002424 012767 002316 176260 MOV #LT1A,SCOLP ;SET SCOPE ADDRESS
1890 002432 004767 020042 JSR PC,LTGER ;GO PRINT LOGIC TEST ERROR
1891 002436 000754 BR LT1C ;CONTINUE TEST
1892 002440 000167 177366 LT1X: JMP TSCD2 ;RETURN TO SCHED
1893

```

```

1894                                     ;LOGIC TEST 2: REGISTER ADDRESSING*****
1895
1896 002444 000240                      LT2:  NOP
1897 002446 012777 000040 176044      LT2IT: MOV  #40,@CS          ;INIT
1898 002454 016777 176144 176036      MOV  DRVN,@CS          ;SELECT DRIVE
1899 002462 012767 027670 176132      MOV  #MSLT2,EMADDR    ;SAVE LT2 HEADER ADDRESS
1900 002470 000240                      NOP
1901 002472 012705 000510              MOV  #C1,R5           ;SET ADDRESS OF FIRST REGISTER
1902 002476 012700 000016              MOV  #16,R0           ;SET NUMBER OF REGISTERS
1903 002502 012702 000626              MOV  #TR00,R2        ;SET START OF REGISTER BUFFER
1904 002506 011501                      LT2A: MOV  (R5),R1
1905 002510 011112                      MOV  (R1),(R2)        ;READ REGISTER
1906 002512 032777 020000 175770      BIT  #20000,@C1       ;SEE IF ERROR
1907 002520 001402                      BEQ  LT2B              ;IF NOT: BR
1908 002522 004767 000032              JSR  PC,LT2ER1        ;ELSE GO TO ERROR 1
1909 002526 000240                      LT2B: NOP
1910 002530 032777 000002 175766      BIT  #2,@ER           ;SEE IF ILR
1911 002536 001402                      BEQ  LT2C              ;IF NOT: BR
1912 002540 004767 000034              JSR  PC,LT2ER2        ;ELSE GO TO ERROR 2
1913 002544 000240                      LT2C: NOP
1914 002546 022225                      CMP  (R2)+,(R5)+     ;BUMP ADDRESS
1915 002550 005300                      DEC  R0
1916 002552 001355                      BNE  LT2A             ;CONTINUE FOR ALL REGISTERS
1917 002554 000240                      NOP
1918 002556 000441                      BR   LT2X
1919 002560 000240                      LT2ER1: NOP
1920 002562 012767 000002 176126      MOV  #2,EXFL          ;FLAG NOT EXPECTED
1921 002570 012767 025644 176074      MOV  #MSG4,ERADD     ;POINT TO CONTROLLER ERROR
1922 002576 000417                      BR   LT2ERG           ;GO TO ERROR
1923 002600 000240                      LT2ER2: NOP
1924 002602 012767 000002 176106      MOV  #2,EXFL          ;FLAG NOT EXPECTED
1925 002610 012767 025662 176054      MOV  #MSG5,ERADD     ;POINT TO DRIVE ERROR
1926 002616 000407                      BR   LT2ERG           ;GO TO ERROR
1927 002620 000240                      LT2ER3: NOP
1928 002622 012767 000001 176066      MOV  #1,EXFL          ;FLAG EXPECTED
1929 002630 012767 025644 176034      MOV  #MSG4,ERADD     ;POINT TO DRIVE
1930 002636 012767 002654 176046      LT2ERG: MOV #LT2LP,SCOLP ;SET SCOPE ADDRESS
1931 002644 004767 017630              JSR  PC,LTGER         ;GO PRINT
1932 002650 000240                      NOP
1933 002652 000207                      RTS  PC               ;ELSE CONTINUE
1934 002654 005726                      LT2LP: TST (SP)+      ;RESET STACK
1935 002656 000167 177624              JMP  LT2A             ;LOOP
1936 002662 000240                      LT2X:  NOP
1937 002664 004767 021232              JSR  PC,ITER          ;GO SEE IF ITERATIONS
1938 002670 000167 177136              JMP  TSCD2            ;RETURN TO SCHED

```



```

1939                                     ;LOGIC TEST 3: CONTROL BUS*****
1940
1941 002674 000240                       LT3:  NOP
1942 002676 012767 027747 175716        LT3IT: MOV #MSLT3,EMADDR ;SET TEST HEADER
1943 002704 012701 000001                MOV #1,R1 ;PRESET PATTERN 1
1944 002710 012700 000020                MOV #20,R0 ;SET PATTERN CHANGE NUMBER
1945 002714 004767 021266                LT3A: JSR PC,INIT1 ;GO INIT
1946 002720 010177 175572                MOV R1,@FC ;WRITE TO FC
1947 002724 032777 000010 175572        BIT #10,@ER ;SEE IF CPAR (TM02)
1948 002732 001013                       BNE LT3ER1 ;IF SO: BR
1949 002734 017702 175556                LT3B: MOV @FC,R2 ;READ FC
1950 002740 032777 020000 175542        BIT #20000,@C1 ;SEE IF MCPE (RH)
1951 002746 001020                       BNE LT3ER2 ;IF SO: BR
1952 002750 005300                       LT3C: DEC R0 ;SEE IF DONE PATTERN CHANGES
1953 002752 001427                       BEQ LT3X ;IF SO: BR
1954 002754 000241                       CLC
1955 002756 006101                       ROL R1 ;CHANGE PATTERN
1956 002760 000755                       BR LT3A ;CONTINUE
1957 002762 012767 026153 175702        LT3ER1: MOV #MSG11,ERADD ;SET ERROR CODE
1958 002770 012767 002714 175714        MOV #LT3A,SCOLP ;SET SCOPE ADDRESS
1959 002776 017702 175514                MOV @FC,R2 ;GET DATA
1960 003002 004767 020640                JSR PC,LTGER1 ;GO DO ERROR
1961 003006 000752                       BR LT3B
1962 003010 012767 026127 175654        LT3ER2: MOV #MSG10,ERADD ;SET ERROR CODE
1963 003016 012767 002734 175666        MOV #LT3B,SCOLP ;SET SCOPE ADDRESS
1964 003024 004767 020616                JSR PC,LTGER1 ;GO DO ERROR
1965 003030 000747                       BR LT3C
1966 003032 105701                       LT3X: TSTB R1 ;SEE IF DONE PATTERN 2
1967 003034 100405                       BMI LT3XX ;IF SO: BR
1968 003036 012701 000401                MOV #401,R1 ;SET PATTERN 2
1969 003042 012700 000010                MOV #10,R0 ;SET PATTERN CHANGE NUMBER
1970 003046 000722                       BR LT3A ;DO PATTERN 2
1971 003050 004767 021046                LT3XX: JSR PC,ITER ;GO SEE IF ITERATIONS
1972 003054 000167 176752                JMP TSCD2 ;RETURN TO SCHEDULAR

```



```

1973
1974
1975
1976 003060 005767 175730
1977 003064 001403
1978 003066 005767 175646
1979 003072 001523
1980 003074 012704 025756
1981 003100 004767 021460
1982 003104 012704 025737
1983 003110 004767 021450
1984 003114 012705 000664
1985 003120 012701 000001
1986 003124 012702 000007
1987 003130 012703 000000
1988 003134 004767 021166
1989 003140 005767 175530
1990 003144 001476
1991 003146 005001
1992 003150 012700 000010
1993 003154 000240
1994 003156 012777 000040 175334
1995 003164 016777 175434 175326
1996 003172 010177 175344
1997 003176 017703 175320
1998 003202 020167 175456
1999 003206 001405
2000 003210 032703 010000
2001 003214 001420
2002 003216 000167 000050
2003 003222 000240
2004 003224 032703 010000
2005 003230 001002
2006 003232 000167 000046
2007 003236 012704 026672
2008 003242 004767 021316
2009 003246 017703 175266
2010 003252 004767 021770
2011 003256 000240
2012 003260 005300
2013 003262 001710
2014 003264 005201
2015 003266 000167 177662
2016 003272 000240
2017 003274 012767 000001 175414
2018 003302 000403
2019 003304 012767 000002 175404
2020 003312 012767 030031 175302
2021 003320 012767 026105 175344
2022 003326 012767 003154 175356
2023 003334 004767 017140
2024 003340 000746
2025 003342 000167 176464
2026

;LOGIC TEST 4: SLAVE ADDRESSING*****
LT4: TST SKAT ;SEE IF SKIP ADDRESS TESTS
      BEQ LT4G0 ;IF NOT: BR
      TST STFLG ;SEE IF SINGLE TEST
      BEQ LT4X ;IF NOT: BR
LT4G0: MOV #MSG8A,R4
      JSR PC,TTOUT ;PRINT TEST INSTRUCTIONS
LT4G: MOV #MSG8,R4
      JSR PC,TTOUT ;REQUEST SLAVE
      MOV #SLVN,R5
      MOV #1,R1
      MOV #7,R2
      MOV #0,R3
      JSR PC,TTR ;GET SLAVE NUMBER
      TST TEMP1 ;SEE IF SLAVE
      BEQ LT4X ;IF NOT: BR
      CLR R1 ;SELECT SLAVE 0
      MOV #10,R0 ;SET NUMBER OF SLAVES
LT4A: NOP
      MOV #40,ACS ;INIT
      MOV DRVN,ACS ;SELECT DRIVE
      MOV R1,ATC ;SELECT SLAVE
      MOV @DS,R3 ;GET DS
      CMP R1,SLVN ;SEE IF SHOULD HAVE SPR
      BEQ LT4B ;IF SO: BR
      BIT #10000,R3 ;SEE IF SPR
      BEQ LT4D ;IF NOT: BR
      JMP LT4ER1 ;GO TO ERROR 1
LT4B: NOP
      BIT #10000,R3 ;SEE IF SPR
      BNE LT4C ;IF SO: BR
      JMP LT4ER2 ;ELSE GO TO ERROR
LT4C: MOV #MSG30,R4 ;PRINT SERIAL NUMBER TAG
      JSR PC,TTOUT
      MOV @SN,R3 ;PRINT SERIAL NUMBER
      JSR PC,SNPT
LT4D: NOP
      DEC R0
      BEQ LT4G ;IF DONE ALL: BR
      INC R1 ;BUMP SLAVE
      JMP LT4A ;CONTINUE
LT4ER1: NOP
      MOV #1,EXFL ;FLAG EXPT: NOT RECEIVED
      BR LT4ERG
LT4ER2: MOV #2,EXFL ;FLAG RECVD: NOT EXPT
LT4ERG: MOV #MSLT4,EMADDR ;SET LT4 HEADER
      MOV #MSG9,ERADD ;SET ERROR CONDITION
      MOV #LT4A,SCOLP ;SET SCOPE ADDRESS
      JSR PC,LTGER ;GO TO ERROR
      BR LT4D ;IF NO SCOPE: BR
LT4X: JMP TSCD2 ;RETURN TO SCHED

```

```

2027                                     ;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST*****
2028
2029 003346 000240                      LT5:  NOP
2030 003350 012767 030110 175244      MOV    #MSLT5,EMADDR      ;SET TEST HEADER
2031 003356 004767 020624              LT5IT: JSR   PC,INIT1      ;GO INIT
2032 003362 012700 000032              MOV    #32,R0             ;SET LOOP FOR BITS 4-0
2033 003366 005001                      CLR    R1                 ;SET TEST WORD
2034 003370 010177 175140              LT5A:  MOV   R1,@MR        ;SEND TEST WORD TO MR
2035 003374 017702 175134              MOV   @MR,R2              ;READ MR
2036 003400 042702 177740              BIC   #177740,R2         ;MASK BITS 4-0
2037 003404 020102                      CMP   R1,R2               ;SEE IF EXPT = RECVD
2038 003406 001402                      BEQ   LT5B                ;IF SO: BR
2039 003410 000167 000064              JMP   LT5ER1             ;ELSE GO TO ERROR 1
2040 003414 000240                      LT5B:  NOP
2041 003416 005300                      DEC   R0                  ;
2042 003420 001402                      BEQ   LT5C                ;IF DONE LOOP: BR
2043 003422 005201                      INC   R1                  ;BUMP TEST WORD
2044 003424 000761                      BR    LT5A                ;CONTINUE LOOP
2045 003426 000240                      LT5C:  NOP
2046 003430 012701 000015              MOV   #15,R1              ;SET TEST WORD + WAM 3
2047 003434 012700 001000              MOV   #1000,R0            ;SET LOOP FOR BITS 15-7
2048 003440 010177 175070              LT5D:  MOV   R1,@MR        ;LOAD MR
2049 003444 017702 175064              MOV   @MR,R2              ;READ MR
2050 003450 042702 000140              BIC   #140,R2             ;MASK OUT BITS 5,6
2051 003454 020102                      CMP   R1,R2               ;SEE IF EXPT = RECVD
2052 003456 001402                      BEQ   LT5E                ;IF SO: BR
2053 003460 000167 000042              JMP   LT5ER2             ;ELSE GO TO ERR 2
2054 003464 000240                      LT5E:  NOP
2055 003466 005300                      DEC   R0                  ;
2056 003470 001431                      BEQ   LT5X                ;IF DONE LOOP: BR
2057 003472 062701 000200              ADD   #200,R1             ;BUMP TEST WORD
2058 003476 000760                      BR    LT5D                ;CONTINUE LOOP
2059 003500 000240                      LT5ER1: NOP
2060 003502 012767 026216 175162      MOV   #MSG14,ERADD        ;SET ERROR CODE
2061 003510 012767 003370 175174      MOV   #LT5A,SCOLP        ;SET SCOPE ADDRESS
2062 003516 004767 020124              JSR   PC,LTGER1          ;GO TO ERROR
2063 003522 000167 177666              JMP   LT5B                ;CONTINUE
2064 003526 000240                      LT5ER2: NOP
2065 003530 012767 026233 175134      MOV   #MSG15,ERADD        ;SET ERROR CODE
2066 003536 012767 003440 175146      MOV   #LT5D,SCOLP        ;SET SCOPE ADDRESS
2067 003544 004767 020076              JSR   PC,LTGER1          ;GO TO ERROR
2068 003550 000167 177710              JMP   LT5E                ;CONTINUE
2069 003554 000240                      LT5X:  NOP
2070 003556 005000                      CLR   R0                  ;
2071 003560 032777 000100 174746      1$:  BIT   #100,@MR        ;TEST FOR 200BPI CLOCK
2072 003566 001012                      BNE   LT5XX              ;
2073 003570 005200                      INC   R0                  ;TRY 65K TIMES
2074 003572 001372                      BNE   1$                 ;
2075 003574 012767 026251 175070      MOV   #MSG15A,ERADD      ;SET ERROR CODE
2076 003602 012767 003554 175102      MOV   #LT5X,SCOLP        ;SET SCOPE ADDRESS
2077 003610 004767 020032              JSR   PC,LTGER1          ;GO TO ERROR
2078 003614 000240                      LT5XX: NOP
2079 003616 004767 020300              JSR   PC,ITER            ;GO SEE IF ITERATIONS
2080 003622 000167 176204              JMP   TSCD2              ;RETURN TO SCHED
2081

```



```

2082                                     ;LOGIC TEST 6: TC REGISTER BIT TEST*****
2083
2084 003626 000240                       LT6:  NOP
2085 003630 012767 030154 174764 LT6IT: MOV  #MSLT6,EMADDR ;POINT TO LT6 HEADER
2086 003636 012700 000003              MOV  #3,R0 ;SET NUMBER OF TESTS
2087 003642 005001                       LT6A1: CLR  R1
2088 003644 004767 020336 LT6A:  JSR  PC,INIT1 ;GO INIT
2089 003650 000240                       LT6B:  NOP
2090 003652 010177 174664              MOV  R1,@TC ;WRITE TC
2091 003656 017702 174660              MOV  @TC,R2 ;READ TC
2092 003662 042702 160000              BIC  #160000,R2 ;MASK OUT TCW
2093 003666 020102                       CMP  R1,R2 ;SEE IF EXPT = RECDV
2094 003670 001402                       BEQ  LT6C ;IF SO: BR
2095 003672 000167 000042              JMP  LT6ER1 ;ELSE GO TO ERROR
2096 003676 000240                       LT6C:  NOP
2097 003700 032777 020000 174634      BIT  #20000,@TC ;SEE IF TCW SET
2098 003706 001002                       BNE  LT6D ;IF SO: BR
2099 003710 000167 000052              JMP  LT6ER2 ;ELSE GO TO ERROR
2100 003714 000240                       LT6D:  NOP
2101 003716 005300                       DEC  R0
2102 003720 001450                       BEQ  LT6X ;IF DONE ALL: BR
2103 003722 022700 000001              CMP  #1,R0 ;SEE IF RESET TEST
2104 003726 001745                       BEQ  LT6A1 ;IF SO: BR
2105 003730 012701 017777              MOV  #17777,R1 ;SET TEST WORD
2106 003734 000167 177704              JMP  LT6A ;DO SET TEST
2107 003740 000240                       LT6ER1: NOP
2108 003742 012767 026323 174722      MOV  #MSG18,ERADD ;SET ERROR CODE
2109 003750 012767 003650 174734      MOV  #LT6B,SCOLP ;SET SCOPE ADDRESS
2110 003756 004767 017664              JSR  PC,LTGER1 ;GO TO ERROR
2111 003762 000167 177710              JMP  LT6C ;CONTINUE
2112 003766 000240                       LT6ER2: NOP
2113 003770 012767 026307 174674      MOV  #MSG17,ERADD ;SET ERROR CODE
2114 003776 010167 174702              MOV  R1,SAV1 ;SAVE R1
2115 004002 012701 000001              MOV  #1,R1 ;SET EXPT = 1
2116 004006 005002                       CLR  R2 ;SET RECDV = 0
2117 004010 012767 004032 174674      MOV  #LT6ER4,SCOLP ;SET SCOPE ADDRESS
2118 004016 004767 017624              JSR  PC,LTGER1 ;GO TO ERROR
2119 004022 016701 174656              MOV  SAV1,R1
2120 004026 000167 177662              JMP  LT6D ;ELSE CONTINUE
2121 004032 005077 174504 LT6ER4: CLR  @TC ;WRITE TO TC
2122 004036 000167 177634              JMP  LT6C ;LOOP ON ERROR
2123 004042 000240                       LT6X:  NOP
2124 004044 004767 020052              JSR  PC,ITER ;GO SEE IF ITERATIONS
2125 004050 000167 175756              JMP  TSCD2 ;RETURN TO SCHED
2126

```



```

2127                                     ;LOGIC TEST 7: FRAME COUNT BIT TEST*****
2128
2129 004054 000240                       LT7:  NOP
2130 004056 012700 000003                LT7IT: MOV    #3,R0           ;SET TEST NUMBER
2131 004062 012767 030220 174532        LT7C:  MOV    #MSLT7,EMADDR ;SET TEST HEADER
2132 004070 005001                       CLR    R1                   ;SET TEST WORD
2133 004072 004767 020110                LT7A:  JSR    PC,INIT1      ;GO INIT
2134 004076 010177 174414                MOV    R1,@FC              ;CLEAR FRAME COUNT
2135 004102 017702 174410                MOV    @FC,R2              ;READ FC
2136 004106 020102                       CMP    R1,R2               ;SEE IF EXPT = RECVD
2137 004110 001402                       BEQ    LT7B                 ;IF SO: BR
2138 004112 000167 000024                JMP    LT7ER1              ;ELSE GO TO ERROR
2139 004116 000240                       LT7B:  NOP
2140 004120 005300                       DEC    R0                   ;SEE IF DONE ALL
2141 004122 001422                       BEQ    LT7X                 ;IF SO: BR
2142 004124 022700 000001                CMP    #1,R0               ;SEE IF RESET TEST
2143 004130 001754                       BEQ    LT7C                 ;IF SO: BR
2144 004132 012701 177777                MOV    #-1,R1              ;SET TEST WORD TO -1
2145 004136 000167 177730                JMP    LT7A                 ;CONTINUE
2146 004142 000240                       LT7ER1: NOP
2147 004144 012767 026342 174520        MOV    #MSG19,ERADD        ;SET ERROR CODE
2148 004152 012767 004072 174532        MOV    #LT7A.SCOLP        ;SET SCOPE ADDRESS
2149 004160 004767 017462                JSR    PC,LTGER1          ;GO PRINT ERROR
2150 004164 000167 177726                JMP    LT7B                 ;ELSE CONTINUE
2151 004170 000240                       LT7X:  NOP
2152 004172 012700 000003                MOV    #3,R0               ;RESET TEST AMT
2153 004176 004767 017720                JSR    PC,ITER             ;GO SEE IF ITERATIONS
2154 004202 000167 175624                JMP    TSCD2               ;RETURN TO SCHED
2155

```

```

2156                                     ;LOGIC TEST 10: FUNCTION CODE BIT TEST*****
2157
2158 004206 000240                       LT10:  NOP
2159 004210 012767 030264 174404        LT10IT: MOV    #MSLT10,EMADDR ;SET TEST HEADER
2160 004216 012700 000003                MOV    #3,R0 ;SET NUMBER OF TESTS
2161 004222 005001                       LT10A1: CLR   R1 ;SET TEST WORRD
2162 004224 012777 000040 174266        LT10A:  MOV    #40,@CS ;INIT
2163 004232 016777 174366 174260        MOV    DRVN,@CS ;SELECT DRIVE
2164 004240 010177 174244                MOV    R1,@C1 ;WRITE C1
2165 004244 017702 174240                MOV    @C1,R2 ;READ C1
2166 004250 042702 177701                BIC    #177701,R2 ;MASK FUNCTION CODE
2167 004254 020102                       CMP    R1,R2 ;SEE IF EXPT = RECVD
2168 004256 001402                       BEQ    LT10B ;IF SO: BR
2169 004260 000167 000024                JMP    LT10E1 ;ELSE GO TO ERROR
2170 004264 000240                       LT10B:  NOP
2171 004266 005300                       DEC    R0
2172 004270 001422                       BEQ    LT10X ;IF DONE ALL: BR
2173 004272 022700 000001                CMP    #1,R0 ;SEE IF RESET TEST
2174 004276 001751                       BEQ    LT10A1 ;IF SO: BR
2175 004300 012701 000076                MOV    #76,R1 ;SET TEST WORD
2176 004304 000167 177714                JMP    LT10A ;DO SET TEST
2177 004310 000240                       LT10E1: NOP
2178 004312 012767 026361 174352        MOV    #MSG20,ERADD ;SET ERROR CODE
2179 004320 012767 004224 174364        MOV    #LT10A,SCOLP ;SET SCOPE ADDRESS
2180 004326 004767 017314                JSR    PC,LTGER1 ;GO PRINT ERROR
2181 004332 000167 177726                JMP    LT10B ;ELSE CONTINUE
2182 004336 000240                       LT10X:  NOP
2183 004340 004767 017556                JSR    PC,ITER ;GO SEE IF ITERATIONS
2184 004344 000167 175462                JMP    TSCD2 ;RETURN TO SCHED

```

```

2185
2186
2187
2188 004350 000240          LT11:  NOP
2189 004352 012767 030337 174242 LT11IT: MOV    #MSLT11,EMADDR ;SET TEST HEADER
2190 004360 004767 017622          JSR    PC,INIT1   ;GO INIT
2191 004364 017702 174120          MOV    @C1,R2     ;READ C1
2192 004370 032702 000001          BIT    #1,R2      ;SEE IF GO=0
2193 004374 001402          BEQ    LT11B      ;IF SO: BR
2194 004376 000167 000072          JMP    LT11E1     ;ELSE GO TO ERROR 1
2195 004402 000240          LT11B: NOP
2196 004404 012777 000015 174122  MOV    #15,@MR    ;SELECT WAM 3
2197 004412 005077 174100          CLR    @FC        ;ASSURE FCS = 1
2198 004416 052777 001700 174116  BIS    #1700,@TC  ;ASSURE FMT OK
2199 004424 012777 000071 174056  MOV    #71,@C1    ;SET READ+GO
2200 004432 017702 174052          MOV    @C1,R2     ;READ C1
2201 004436 032702 000001          BIT    #1,R2      ;SEE IF GO =1
2202 004442 001002          BNE    LT11C      ;IF SO: BR
2203 004444 000167 000060          JMP    LT11E2     ;ELSE GO TO ERROR 2
2204 004450 000240          LT11C: NOP
2205 004452 004767 017530          JSR    PC,INIT1   ;GO INIT
2206 004456 017702 174026          MOV    @C1,R2     ;READ C1
2207 004462 032702 000001          BIT    #1,R2      ;SEE IF GO=0
2208 004466 001452          BEQ    LT11X      ;IF SO:BR
2209 004470 000167 000070          JMP    LT11E3     ;ELSE GO TO ERROR 3
2210 004474 000240          LT11E1: NOP
2211 004476 012767 026413 174166  MOV    #MSG21,ERADD ;SET ERROR CODE
2212 004504 012702 000001          MOV    #1,R2      ;SET REVD
2213 004510 005001          CLR    R1         ;SET EXPT
2214 004512 012767 004352 174172  MOV    #LT11IT,SCOLP ;SET SCOPE ADDRESS
2215 004520 004767 017122          JSR    PC,LTGER1  ;GO PRINT ERROR
2216 004524 000167 177652          JMP    LT11B      ;ELSE CONTINUE
2217 004530 000240          LT11E2: NOP
2218 004532 012767 026451 174132  MOV    #MSG22,ERADD ;SET ERROR CODE
2219 004540 005002          CLR    R2         ;SET RCVD
2220 004542 012701 000001          MOV    #1,R1      ;SET EXPT
2221 004546 012767 004402 174136  MOV    #LT11B,SCOLP ;SET SCOPE ADDRESS
2222 004554 004767 017066          JSR    PC,LTGER1  ;GO PRINT ERROR
2223 004560 000167 177664          JMP    LT11C      ;ELSE CONTINUE
2224 004564 000240          LT11E3: NOP
2225 004566 012767 026472 174076  MOV    #MSG23,ERADD ;SET ERROR CODE
2226 004574 005001          CLR    R1         ;SET EXPT
2227 004576 012702 000001          MOV    #1,R2      ;SET RCVD
2228 004602 012767 004450 174102  MOV    #LT11C,SCOLP ;SET SCOPE ADDRESS
2229 004610 004767 017032          JSR    PC,LTGER1  ;GO PRINT ERROR
2230 004614 000240          LT11X: NOP
2231 004616 004767 017300          JSR    PC,ITER    ;GO SEE IF ITERATIONS
2232 004622 000167 175204          JMP    TSCD2      ;RETURN TO SCHED

```



```

2233
2234           ;LOGIC TEST 12: DRIVE READY BIT*****
2235
2236 004626 000240          LT12:  NOP
2237 004630 012767 030404 173764 LT12IT: MOV    #MSLT12,EMADDR ;SET TEST HEADER
2238 004636 004767 017344          JSR    PC,INIT1   ;GO INIT
2239 004642 032777 000200 173652   BIT    #200,ADS   ;SEE IF DRY=1
2240 004650 001002          BNE    LT12B      ;IF SO: BR
2241 004652 000167 000066          JMP    LT12E1     ;ELSE GO TO ERROR 1
2242 004656 000240          LT12B: NOP
2243 004660 012777 000015 173646   MOV    #15,@MR    ;SET WAM3
2244 004666 005077 173624          CLR    @FC        ;ASSURE FCS = 1
2245 004672 052777 001700 173642   BIS    #1700,@TC  ;ASSURE FMT OK
2246 004700 012777 000071 173602   MOV    #71,@C1    ;SET READ+GO
2247 004706 032777 000200 173606   BIT    #200,ADS   ;SEE IF DRY=0
2248 004714 001402          BEQ    LT12C      ;IF SO: BR
2249 004716 000167 000046          JMP    LT12E2     ;ELSE GO TO ERROR 2
2250 004722 000240          LT12C: NOP
2251 004724 004767 017256          JSR    PC,INIT1   ;GO INIT
2252 004730 032777 000200 173564   BIT    #200,ADS   ;SEE IF DRY=1
2253 004736 001036          BNE    LT12X      ;IF SO: BR
2254 004740 000167 000050          JMP    LT12E3     ;ELSE GO TO ERROR 3
2255 004744 000240          LT12E1: NOP
2256 004746 012767 026525 173716   MOV    #MSG24,ERADD ;SET ERROR CODE
2257 004754 012767 004630 173730   MOV    #LT12IT,SCOLP ;SET SCOPE ADDRESS
2258 004762 004767 016652          JSR    PC,LTGER2  ;GO TO ERROR
2259 004766 000733          BR     LT12B      ;CONTINUE
2260 004770 000240          LT12E2: NOP
2261 004772 012767 026553 173672   MOV    #MSG25,ERADD ;SET ERROR CODE
2262 005000 012767 004656 173704   MOV    #LT12B,SCOLP ;SET LOOP ADDRESS
2263 005006 004767 016626          JSR    PC,LTGER2  ;GO PRINT ERROR
2264 005012 000743          BR     LT12C      ;CONTINUE
2265 005014 012767 026602 173650   LT12E3: MOV    #MSG25A,ERADD ;SET ERROR CODE
2266 005022 012767 004722 173662   MOV    #LT12C,SCOLP ;SET ERROR LOOP
2267 005030 004767 016604          JSR    PC,LTGER2  ;GET PRINT ERROR
2268 005034 000240          LT12X: NOP
2269 005036 004767 017060          JSR    PC,ITER    ;GO TO ITERATION SUBROUTINE
2270 005042 000167 174764          JMP    TSCD2      ;RETURN TO SCHED

```

```
2271
2272
2273
2274 005046 000240
2275 005050 005000
2276 005052 012767 030455 173542
2277 005060 004767 017122
2278 005064 012767 005146 173576
2279 005072 005077 173412
2280 005076 005077 173464
2281 005102 052777 000100 173400
2282 005110 000240
2283 005112 005300
2284 005114 001376
2285 005116 000240
2286 005120 012777 000340 173440
2287 005126 012767 026627 173536
2288 005134 012767 005060 173550
2289 005142 004767 016472
2290 005146 000240
2291 005150 004767 016746
2292 005154 000167 174652

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

2293  
2294  
2295  
2296  
2297  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312  
2313  
2314  
2315  
2316  
2317  
2318  
2319  
2320  
2321  
2322  
2323  
2324  
2325  
2326

005160 000240  
005162 032777 001000 173400  
005170 001005  
005172 005767 173542  
005176 001437  
005200 000167 016762  
005204 012767 030522 173410  
005212 012704 033443  
005216 004767 017012  
005222 004767 016760  
005226 012701 014602  
005232 017702 173264  
005236 004767 000040  
005242 020102  
005244 001411  
005246 000240  
005250 012767 005222 173434  
005256 012767 026656 173406  
005264 004767 016356  
005270 000240  
005272 004767 016624  
005276 000167 174530  
005302 032702 000040  
005306 001402  
005310 052701 000040  
005314 000207

```

;THE NEXT 4 TESTS ARE MANUAL INTERVENTION STATUS TESTS.
;THE OPERATOR WILL BE REQUIRED TO MANIPULATE THE TU45
;CONTROL PANEL IN ACCORDANCE WITH TTY INSTRUCTIONS.

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

LT14:  NOP
      BIT    #1000,@SWR      ;SEE IF INHIB MAN TST
      BNE   LT14A           ;IF NOT: BR
      TST   STFLG           ;SEE IF SINGLE TEST
      BEQ   LT14XX          ;IF NOT: BR
      JMP   INMT            ;ELSE GO PRINT INHIB MSG
LT14A: MOV   #MSLT14,EMADDR ;SET TEST HEADER
      MOV   #MSG1,R4        ;SET INSTRUCTION ONE
      JSR   PC,INST         ;GO DO INSTRUCTION
LT14IT: JSR  PC,INIT1       ;INIT, SELECT DRIVE + SLAVE
      MOV   #14602,R1       ;SET TEST WORD
      MOV   @DS,R2          ;ASSURE MOL,WRL,DPR,DRY,BOT
      JSR   PC,SETPES
      CMP   R1,R2
      BEQ   LT14X           ;IF SO: BR
      NOP
      MOV   #LT14IT,SCOLP   ;SET LOOP ADDRESS
      MOV   #MSG27,ERADD    ;SET ERROR CODE
      JSR   PC,LTGER1       ;GO PRINT ERROR
LT14X: NOP
      JSR   PC,ITER         ;GO SEE IF ITERATION
LT14XX: JMP  TSCD2         ;RETURN TO SCHED

SETPES: BIT    #40,R2
        BEQ   1$
        BIS   #40,R1
1$:     RTS    PC
    
```



```

2327
2328                                     ;LOGIC TEST 15: STATUS AT BOT, OFFLINE, LOADED, NO WRITE RING*****
2329
2330 005316 000240          LT15:  NOP
2331 005320 032777 001000 173242  BIT      #1000,@SWR      ;SEE IF INHIB MAN TST
2332 005326 001005          BNE      LT15A        ;IF NOT: BR
2333 005330 005767 173404    TST      STFLG        ;SEE IF SINGLE TEST
2334 005334 001437          BEQ      LT15XX       ;IF NOT: BR
2335 005336 000167 016624    JMP      INMT         ;ELSE GO PRINT INHIB MSG
2336 005342 012767 030570 173252  LT15A:  MOV      #MSLT15,EMADDR ;SET TEST HEADER
2337 005350 012704 033541    MOV      #MMSG2,R4
2338 005354 004767 016654    JSR      PC,INST     ;PRINT INSTRUCTION
2339 005360 004767 016632    LT15IT: JSR      PC,INIT2  ;GO INIT, SELECT DRIVE, SLAV
2340 005364 012701 100700    MOV      #100700,R1  ;SET TEST WORD
2341 005370 017702 173126    MOV      @DS,R2      ;READ STATUS
2342 005374 004767 177702    JSR      PC,SETPES
2343 005400 020102          CMP      R1,R2       ;SEE OF EXPT=RCVD
2344 005402 001411          BEQ      LT15X
2345 005404 000240          NOP
2346 005406 012767 005360 173276  MOV      #LT15IT,SCOLP ;SET LOOP ADDRESS
2347 005414 012767 026656 173250  MOV      #MSG27,ERADD ;SET ERROR CODE
2348 005422 004767 016220    JSR      PC,LTGER1  ;GO PRINT ERROR
2349 005426 000240          LT15X:  NOP
2350 005430 004767 016466    JSR      PC,ITER     ;GO SEE IF ITERATIONS
2351 005434 000167 174372    LT15XX: JMP      TSCD2     ;RETURN TO SCHED

```

```

2352
2353                                     ;LOGIC TEST 16: STATUS AT EOT, OFFLINE LOADED, NO WRITE RING*****
2354
2355 005440 000240          LT16:  NOP
2356 005442 032777 001000 173120  BIT      #1000,@SWR      ;SEE IF INHIB MAN TST
2357 005450 001005          BNE      LT16A      ;IF NOT: BR
2358 005452 005767 173262  TST      STFLG      ;SEE IF SINGLE TEST
2359 005456 001437          BEQ      LT16XX     ;IF NOT: BR
2360 005460 000167 016502  JMP      INMT       ;ELSE GO PRINT INHIB MSG
2361 005464 012767 030636 173130  LT16A:  MOV      #MSLT16,EMADDR ;SET TEST HEADER
2362 005472 012704 033562  MOV      #MMSG3,R4
2363 005476 004767 016532  JSR      PC,INST    ;GO PRINT INSTRUCTION
2364 005502 004767 016510  LT16IT: JSR      PC,INIT2 ;SELECT DRIVE,SLAVE
2365 005506 012701 116701  MOV      #116701,R1 ;SET TEST WORD
2366 005512 017702 173004  MOV      @DS,R2     ;READ STATUS
2367 005516 004767 177560  JSR      PC,SETPES
2368 005522 020102          CMP      R1,R2      ;SEE IF EXPT=RCVD
2369 005524 001411          BEQ      LT16X     ;IF SO: BR
2370 005526 000240          NOP
2371 005530 012767 005502 173154  MOV      #LT16IT,SCOLP ;SET LOOP ADDRESS
2372 005536 012767 026656 173126  MOV      #MSG27,ERADD ;SET ERROR CODE
2373 005544 004767 016076  JSR      PC,LTGER1  ;GO PRINT ERROR
2374 005550 000240          LT16X:  NOP
2375 005552 004767 016344  JSR      PC,ITER    ;GO SEE IF ITERATION
2376 005556 000167 174250  LT16XX: JMP      TSCD2  ;RETURN TO SCHED
2377

```

```

2378
2379                                     ;LOGIC TEST 17: STATUS AT ON LINE, LOADED*****
2380
2381 005562 000240          LT17:  NOP
2382 005564 032777 001000 172776  BIT      #1000,@SWR      ;SEE IF INHIB MAN TST
2383 005572 001005          BNE      LT17A      ;IF NOT: BR
2384 005574 005767 173140  TST      STFLG      ;SEE IF SINGLE TEST
2385 005600 001444          BEQ      LT17XX     ;IF NOT: BR
2386 005602 000167 016360  JMP      INMT       ;ELSE GO PRINT INHIB MSG
2387 005606 012767 030704 173006  LT17A:  MOV      #MSLT17,EMADDR ;SET TEST HEADER
2388 005614 012704 033620  MOV      #MMSG4,R4
2389 005620 004767 016410  JSR      PC,INST    ;GO PRINT INSTRUCTION
2390 005624 004767 016366  LT17IT: JSR      PC,INIT2 ;SELECT DRIVE, SLAVE
2391 005630 012701 112701  MOV      #112701,R1 ;SET TEST WORD
2392 005634 017702 172662  MOV      @DS,R2    ;READ STATUS
2393 005640 004767 177436  JSR      PC,SETPES
2394 005644 032702 002000  BIT      #2000,R2
2395 005650 001002          BNE      1$
2396 005652 042701 002000  BIC      #2000,R1
2397 005656 020102          1$:  CMP      R1,R2      ;SEE IF EXPT=RCVD
2398 005660 001411          BEQ      LT17X     ;IF SO: BR
2399 005662 000240          NOP
2400 005664 012767 005624 173020  MOV      #LT17IT,SCOLP ;SET LOOP ADDRESS
2401 005672 012767 026656 172772  MOV      #MSG27,ERADD ;SET ERROR CODE
2402 005700 004767 015742  JSR      PC,LTGER1  ;YES PRINT ERROR
2403 005704 000240          LT17X:  NOP
2404 005706 004767 016210  JSR      PC,ITER    ;GO SEE IF ITERATIONS
2405 005712 000167 174114  LT17XX: JMP      TSCD2 ;RETURN TO SCHED

```



```

2406
2407
2408
2409
2410
2411
2412
2413
2414 005716 000240
2415 005720 012767 030752 172674
2416 005726 012767 005746 172756
2417 005734 012700 000022
2418 005740 012767 000544 172726
2419 005746 004767 016234
2420 005752 012777 177777 172532
2421 005760 012701 000001
2422 005764 117777 172704 172516
2423 005772 017702 172526
2424 005776 030102
2425 006000 001011
2426 006002 012767 034103 172662
2427 006010 012767 000001 172700
2428 006016 004767 014450
2429 006022 000405
2430 006024 000240
2431 006026 020102
2432 006030 001402
2433 006032 004767 014422
2434 006036 005300
2435 006040 001404
2436 006042 005267 172626
2437 006046 000167 177674
2438 006052 000240
2439 006054 004767 016042
2440 006060 004767 015066
2441 006064 000167 173742

```

```

;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 AVAILABL 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)*****

LT20:  NOP
      MOV #MSLT20,EMADDR ;SET TEST HEADER
      MOV #LT20A,SCOLP ;SET LOOP ADDRESS
LT20IT: MOV #22,R0 ;SET NUMBER OF ILL CODES
      MOV #ILFT,TEMP1 ;POINT TO START IF TABLE
LT20A: JSR PC,INIT1 ;GO INIT, SELECT SLAVE + DRIVE
      MOV #-1,@WC ;SET WC= -1
      MOV #1,R1 ;SET TEST WORD
      MOVB @TEMP1,@C1 ;SET ILL CODE
      MOV @ER,R2 ;READ ER
      BIT R1,R2 ;SEE IF EXPT=RCVD
      BNE LT20B ;IF SO: BR
      MOV #TMS17,ERADD ;SET ERROR CODE
      MOV #1,EXFL ;SET EXPT FLG
      JSR PC,LTGER0 ;GO PRINT ERROR
      BR LT20C
LT20B: NOP
      CMP R1,R2 ;SEE UNEXPECTED ERRORS
      BEQ LT20C ;IF NOT: BR
      JSR PC,LTGER3 ;ELSE PRINT ERROR
LT20C: DEC R0 ;SEE IF DONE ALL ILL CODES
      BEQ LT20X ;IF SO: BR
      INC TEMP1 ;BUMP ADDRESS
      JMP LT20A ;CONTINUE
LT20X: NOP
      JSR PC,ITER ;GO SEE IF ITERATION
      JSR PC,DRVCLR
      JMP TSCD2 ;RETURN TO SCHED

```

```

2442
2443
2444
2445 006070 000240
2446 006072 012767 031031 172522
2447 006100 012767 006106 172604
2448 006106 004767 016074
2449 006112 052777 000300 172422
2450 006120 012777 000015 172406
2451 006126 012777 000071 172354
2452 006134 005077 172356
2453 006140 012701 000004
2454 006144 017702 172354
2455 006150 030102
2456 006152 001011
2457 006154 012767 034117 172510
2458 006162 012767 000001 172526
2459 006170 004767 014276
2460 006174 000405
2461 006176 000240
2462 006200 020102
2463 006202 001402
2464 006204 004767 014250
2465 006210 004767 015706
2466 006214 012703 040000
2467 006220 005303
2468 006222 001376
2469 006224 004767 014026
2470 006230 004767 014716
2471 006234 000167 173572

;LOGIC TEST 21: REGISTER MODIFICATION REFUSED(RMR)*****
LT21: NOP
MOV #MSLT21,EMADDR ;SET TEST HEADER
MOV #LT21IT,SCOLP ;SET SCOPE LOOP ADDRESS
LT21IT: JSR PC,INIT1 ;GO INIT, SELECT SLAVE, DRIVE
BIS #300,@TC ;SET FORMAT
MOV #15,@MR ;SET WAM3
MOV #71,@C1 ;SET READ+GO
CLR @FC ;ATTEMPT WRITE TO FC
MOV #4,R1 ;SET TEST WORD
MOV @ER,R2 ;GET ER
BIT R1,R2 ;SEE IF EXPT=RCVD
BNE LT21A ;IF SO: BR
MOV #TMS19,ERADD ;SET ERROR CODE
MOV #1,EXFL ;SET EXPT FLG
JSR PC,LTGERO ;GO PRINT ERROR
BR LT21B
LT21A: NOP
CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
BEQ LT21B ;IF NOT: BR
JSR PC,LTGER3 ;ELSE GO PRINT ERROR
LT21B: JSR PC,ITER ;GO SEE IF ITERATION
MOV #40000,R3
LT21XA: DEC R3 ;DELAY FOR ALPHA
BNE LT21XA
JSR PC,EORPA ;GO DO EOR CLEAR
JSR PC,DRVCLR
JMP TSCD2 ;RETURN TO SCHED

```

```

2472
2473
2474
2475 006240 000240
2476 006242 012767 031065 172352
2477 006250 012767 006256 172434
2478 006256 004767 015724
2479 006262 052777 000020 172230
2480 006270 012777 177777 172220
2481 006276 000240
2482 006300 012701 000010
2483 006304 042777 000020 172206
2484 006312 017702 172206
2485 006316 030102
2486 006320 001011
2487 006322 012767 034125 172342
2488 006330 012767 000001 172360
2489 006336 004767 014130
2490 006342 000405
2491 006344 000240
2492 006346 020102
2493 006350 001402
2494 006352 004767 014102
2495 006356 004767 015540
2496 006362 004767 014564
2497 006366 000167 173440

;LOGIC TEST 22: CONTROL BUS PARITY (CPAR)*****
LT22:  NOP
      MOV #MSLT22,EMADDR ;SET TEST HEADER
      MOV #LT22IT,SCOLP  ;SET SCOPE LOOP ADDRESS
LT22IT: JSR PC,INIT1     ;INIT, SELECT SLAVE+DRIVE
      BIS #20,ACS        ;ENABLE EVEN PARITY ON MB
      MOV #-1,@FC        ;WRITE TO FC
      NOP
      MOV #10,R1         ;SET TEST WORD
      BIC #20,ACS        ;RESET PARITY TO ODD
      MOV @ER,R2         ;GET ER
      BIT R1,R2          ;SEE IF EXPT=RCVD
      BNE LT22A          ;IF SO: BR
      MOV #TMS20,ERADD   ;SET ERROR CODE
      MOV #1,EXFL        ;SET EXPT FLG
      JSR PC,LTGER0      ;GO PRINT ERROR
      BR LT22X
LT22A: NOP
      CMP R1,R2          ;SEE IF UNEXPECTED ERRORS
      BEQ LT22X          ;IF NOT: BR
      JSR PC,LTGER3      ;ELSE GO PRINT ERROR
LT22X: JSR PC,ITER       ;GO SEE IF ITERATION
      JSR PC,DRVCLR
      JMP TSCD2          ;RETURN TO SCHED

```



```

2498
2499
2500           ;LOGIC TEST 23: FORMAT ERROR(FMT)*****
2501 006372 000240          LT23:  NOP
2502 006374 012767 031122 172220  MOV    #MSLT23,EMADDR ;SET TEST HEADER
2503 006402 012767 006410 172302  MOV    #LT23IT,SCOLP  ;SET SCOPE ADDRESS
2504 006410 004767 015572          LT23IT: JSR    PC,INIT1    ;GO INIT SELECT DRIVE+SLAVE
2505 006414 042777 000360 172120  BIC    #360,@TC      ;SET ILLEGAL FORMAT
2506 006422 012701 000020          MOV    #20,R1        ;SET TEST WORD
2507 006426 012777 000015 172100  MOV    #15,@MR       ;SET WAM 3
2508 006434 012777 000071 172046  MOV    #71,@C1       ;SET READ+GO
2509 006442 017702 172056          MOV    @ER,R2        ;READ ER
2510 006446 030102          BIT    R1,R2        ;SEE IF EXPT=RCVD
2511 006450 001011          BNE    LT23A        ;IF SO: BR
2512 006452 012767 034134 172212  MOV    #TMS21,ERADD  ;SET ERROR CODE
2513 006460 012767 000001 172230  MOV    #1,EXFL       ;SET EXPT FLG
2514 006466 004767 014000          JSR    PC,LTGERO    ;GO PRINT ERROR
2515 006472 000404          BR     LT23X
2516 006474 020102          LT23A: CMP    R1,R2    ;SEE IF UNEXPECTED ERRORS
2517 006476 001402          BEQ    LT23X       ;IF NOT: BR
2518 006500 004767 013754          JSR    PC,LTGER3   ;ELSE GO PRINT ERROR
2519 006504 000240          LT23X: NOP
2520 006506 004767 015410          JSR    PC,ITER     ;GO SEE IF ITERATION
2521 006512 004767 013540          JSR    PC,EORPA
2522 006516 004767 014430          JSR    PC,DRVCLR
2523 006522 000167 173304          JMP    TSCD2      ;RETURN TO SCHED

```

```

2524                                     ;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)*****
2525
2526 006526 012767 031164 172066 LT24:  MOV    #MSLT24,EMADDR  ;SET TEST HEADER
2527 006534 012767 006542 172150          MOV    #LT24IT,SCOLP  ;SET SCOPE ADDRESS
2528 006542 012767 000005 172036 LT24IT: MOV    #5,ITAMT
2529 006550 004767 015432          JSR    PC,INIT1      ;GO INIT, SELECT DRIVE+SLAVE
2530 006554 052777 000300 171760          BIS    #300,@TC      ;SET NORMAL FORMAT
2531 006562 012777 034350 171724          MOV    #WDATA,@BA    ;SET BA
2532 006570 012777 177760 171720          MOV    #-20,@FC      ;SET FC
2533 006576 012777 177770 171706          MOV    #-10,@WC      ;SET WC
2534 006604 012777 000013 171722          MOV    #13,@MR       ;SELECT WAM 2
2535 006612 012777 000061 171670          MOV    #61,@C1       ;SET WRITE+GO
2536 006620 052777 000020 171672          BIS    #20,@CS       ;FORCE EVEN PARITY
2537 006626 012701 000040          MOV    #40,R1        ;SET TEST WORD
2538 006632 012703 000004          MOV    #4,R3
2539 006636 005000          CLR    R0
2540 006640 005300          LT24A: DEC    R0
2541 006642 001376          BNE    LT24A         ;DELAY
2542 006644 005303          DEC    R3
2543 006646 001374          BNE    LT24A
2544 006650 012700 000004          MOV    #4,R0
2545 006654 012777 000013 171652 LT24B: MOV    #13,@MR       ;CLOCK MR 4 TIMES
2546 006662 005300          DEC    R0
2547 006664 022700 000002          CMP    #2,R0         ;SEE IF DONE 1 BYTE
2548 006670 001002          BNE    LT24B0        ;IF NOT: BR
2549 006672 017701 171636          MOV    @MR,R1        ;ELSE GET BYTE 1
2550 006676 005700          LT24B0: TST   R0      ;SEE IF BYTE 2
2551 006700 001365          BNE    LT24B         ;IF NOT: BR
2552 006702 017704 171626          MOV    @MR,R4        ;GET BYTE 2
2553 006706 005000          CLR    R0
2554 006710 005300          LT24C: DEC    R0
2555 006712 001376          BNE    LT24C         ;DELAY
2556 006714 032777 000040 171602          BIT    #40,@ER       ;SEE IF DPAR IS SET
2557 006722 001023          BNE    LT24D         ;IF SO: BR
2558 006724 000301          SWAB   R1
2559 006726 042701 177400          BIC    #177400,R1    ;GET LOW BYTE
2560 006732 042704 000377          BIC    #377,R4
2561 006736 050401          BIS    R4,R1         ;GET HIGH BYTE
2562 006740 005267 172000          INC    T24FL         ;SET T24 FLAG
2563 006744 012767 034142 171720          MOV    #TMS22,ERADD ;SET ERROR CODE
2564 006752 012767 000001 171736          MOV    #1,EXFL       ;SET EXPT FLG
2565 006760 004767 013506          JSR    PC,LTGERO     ;GO PRINT ERROR
2566 006764 005067 171754          CLR    T24FL        ;CLEAR FLAG
2567 006770 000412          BR     LT24X
2568 006772 012701 000050          LT24D: MOV    #50,R1
2569 006776 017702 171522          MOV    @ER,R2        ;GET ERROR REGISTER
2570 007002 042702 020000          BIC    #20000,R2    ;MASK OPI
2571 007006 020102          CMP    R1,R2        ;SEE IF UNEXPECTED ERRORS
2572 007010 001402          BEQ    LT24X        ;IF NOT: BR
2573 007012 004767 013442          JSR    PC,LTGER3     ;ELSE GO PRINT ERROR
2574 007016 042777 000020 171474 LT24X: BIC    #20,@CS       ;RESET EVEN PARITY
2575 007024 004767 013226          JSR    PC,EORPA      ;GO DO EOR CLEAR
2576 007030 004767 014116          JSR    PC,DRVCLR     ;GO SEE IF DRIVE CLEAR OK
2577 007034 004767 015062          JSR    PC,ITER       ;GO SEE IF ITERATION
2578 007040 012767 000020 171540          MOV    #20,ITAMT
2579 007046 000167 172760          JMP    TSCD2        ;RETURN TO SCHED

```



```

2580
2581
2582
2583 007052 012767 031224 171542 LT25: MOV #MSLT25,EMADDR ;SET TEST HEADER
2584 007060 004767 015122 LT25IT: JSR PC,INIT1 ;INIT, SELECT DRIVE+SLAVE
2585 007064 052777 000300 171450 BIS #300,@TC ;SET NORMAL FORMAT
2586 007072 012777 177777 171416 MOV #-1,@FC ;SET ITLLEGAL FC
2587 007100 012777 000013 171426 MOV #13,@MR ;SET WAM 2
2588 007106 012777 000061 171374 MOV #61,@C1 ;LOAD WRITE+GO
2589 007114 012701 004000 MOV #4000,R1 ;SET TEST WORD
2590 007120 017702 171400 MOV @ER,R2 ;GET ER
2591 007124 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2592 007126 001014 BNE LT25A ;IF SO: BR
2593 007130 012767 007060 171554 MOV #LT25IT,SCOLP ;SET LOOP ADDRESS
2594 007136 012767 034230 171526 MOV #TMS31,ERADD ;SET ERROR CODE
2595 007144 012767 000001 171544 MOV #1,EXFL ;SET EXPT FLAG
2596 007152 004767 013314 JSR PC,LTGERO ;GO PRINT ERROR
2597 007156 000404 BR LT25X
2598 007160 020102 LT25A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2599 007162 001402 BEQ LT25X ;IF NOT: BR
2600 007164 004767 013270 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2601 007170 004767 014726 LT25X: JSR PC,ITER ;GO SEE IF ITERATION
2602 007174 004767 013752 JSR PC,DRVCLR
2603 007200 000167 172626 JMP TSCD2 ;RETURN TO SCHED
  
```



```

2604
2605
2606
2607 007204 012767 031260 171410 LT26: MOV #MSLT26,EMADDR ;SET TEST HEADER
2608 007212 004767 014770 LT26IT: JSR PC,INIT1 ;INIT, SELECT DRIVE+SLAVE
2609 007216 005000 CLR R0
2610 007220 005300 LT26W: DEC R0
2611 007222 001376 BNE LT26W ;AWAIT OPI RESET
2612 007224 052777 000300 171310 BIS #300,@TC ;SET NORMAL FORMAT
2613 007232 012777 177770 171252 MOV #-10,@WC ;SET WC=-10
2614 007240 012777 177760 171250 MOV #-20,@FC ;SET FC=-20
2615 007246 012777 000013 171260 MOV #13,@MR ;SET WAM 3
2616 007254 012777 000061 171226 MOV #61,@C1 ;LOAD WRITE+GO
2617 007262 012701 001000 MOV #1000,R1 ;SET TEST WORD
2618 007266 005000 CLR R0
2619 007270 005300 LT26A: DEC R0
2620 007272 001376 BNE LT26A ;DELAY
2621 007274 012777 000025 171232 MOV #25,@MR ;LOAD MM EOR CLEAR
2622 007302 105077 171226 CLRB @MR ;RESET MR
2623 007306 012703 000004 MOV #4,R3
2624 007312 005000 CLR R0
2625 007314 032777 001000 171202 LT26B: BIT #1000,@ER ;SEE IF FCE SET
2626 007322 001022 BNE LT26C ;IF SO: BR
2627 007324 005300 DEC R0
2628 007326 001372 BNE LT26B ;DELAY
2629 007330 005303 DEC R3
2630 007332 001370 BNE LT26B
2631 007334 017702 171164 MOV @ER,R2 ;GET ER
2632 007340 012767 007212 171344 MOV #LT26IT,SCOLP ;SET SCOPE ADDRESS
2633 007346 012767 034207 171316 MOV #TMS28,ERADD
2634 007354 012767 000001 171334 MOV #1,EXFL ;SET EXPT FLG
2635 007362 004767 013104 JSR PC,LTGERO ;GO PRINT ERROR
2636 007366 000406 BR LT26X
2637 007370 017702 171130 LT26C: MOV @ER,R2 ;GET ERROR REGISTER
2638 007374 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2639 007376 001402 BEQ LT26X ;IF NOT: BR
2640 007400 004767 013054 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2641 007404 004767 014512 LT26X: JSR PC,ITER ;GO SEE IF ITERATION
2642 007410 004767 013536 JSR PC,DRVCLR
2643 007414 000167 172412 JMP TSCD2 ;RETURN TO SCHED
    
```

```

2644
2645
2646
2647 007420 022767 172400 171062 LT27:  CMP      #172400,C1      ;SEE IF ADDRESSES OPEN
2648 007426 001041                BNE      LT27XX      ;IF NOT: BR
2649 007430 012767 007454 171254        MOV      #LT27A,SCOLP ;SET SCOPE ADDRESS
2650 007436 012767 031314 171156        MOV      #MSLT27,EMADDR ;SET TEST HEADER
2651 007444 012700 000020                LT27IT: MOV      #20,R0      ;SET NUMBER OF ILR TESTS
2652 007450 012701 172434                MOV      #172434,R1    ;SET FIRST ILR ADDRESS
2653 007454 004767 014526                LT27A:  JSR      PC,INIT1 ;GO INIT, SELECT DRIVE+SLAVE
2654 007460 011103                MOV      (R1),R3      ;ATTEMPT ILR READ
2655 007462 032777 000002 171034        BIT      #2,@ER      ;SEE IF ILR=1
2656 007470 001010                BNE      LT27B      ;IF SO: BR
2657 007472 012767 000001 171216        MOV      #1,EXFL     ;SET EXPT-NOT RCVD FLAG
2658 007500 012767 034031 171164        MOV      #TMS10,ERADD ;SET ERROR CODE
2659 007506 004767 012766                JSR      PC,LTGER     ;GO PRINT ERROR
2660 007512 005300                LT27B:  DEC      R0      ;SEE IF DONE ALL
2661 007514 001402                BEQ      LT27X      ;IF SO: BR
2662 007516 005721                TST      (R1)+      ;BUMP ADDRESS
2663 007520 000755                BR       LT27A      ;CONTINUE TESTS
2664 007522 004767 014374                LT27X:  JSR      PC,ITER ;GO SEE IF ITERATIONS
2665 007526 004767 013420                JSR      PC,DRVCLR
2666 007532 000167 172274                LT27XX: JMP      TSCD2   ;RETURN TO SCHED

```



```

2667
2668
2669
2670 007536 012767 034236 171126 LT30: MOV #TMS32,ERADD ;SET ERROR CODE
2671 007544 012767 031350 171050 MOV #MSLT30,EMADDR ;SET TEST HEADER
2672 007552 012767 007560 171132 MOV #LT30IT,SCOLP ;SET SCOPE ADDRESS
2673 007560 004767 014422 LT30IT: JSR PC,INIT1 ;INIT, SELECT DRIVE + SLAVE
2674 007564 052777 000300 170750 BIS #300,@TC ;SET NORMAL FORMAT
2675 007572 012701 010000 MOV #10000,R1 ;SET TEST WORD
2676 007576 012777 000017 170730 MOV #17,@MR ;CRIPPLE OCCUPIED
2677 007604 005077 170706 CLR @FC ;SET FC3
2678 007610 012777 000061 170672 MOV #61,@C1 ;LOAD WRITE+GO
2679 007616 032777 010000 170700 BIT #10000,@ER ;SEE IF DTE SET
2680 007624 001005 BNE LT30A ;IF SO: BR
2681 007626 012767 000001 171062 MOV #1,EXFL ;SET EXPT FLG
2682 007634 004767 012632 JSR PC,LTGER0 ;GO PRINT ERROR
2683 007640 004767 014342 LT30A: JSR PC,INIT1 ;GO INIT SELECT DRIVE,SLAVE
2684 007644 052777 000300 170670 BIS #300,@TC ;SET FORMAT
2685 007652 012701 010000 MOV #10000,R1 ;SET TEST WORD
2686 007656 005077 170634 CLR @FC ;SET FCS
2687 007662 012777 000015 170644 MOV #15,@MR ;SET WRAP 3
2688 007670 012777 000061 170612 MOV #61,@C1 ;LOAD WRITE+GO
2689 007676 012704 040000 MOV #40000,R4
2690 007702 005777 170634 LT30B: TST @TC ;SEE IF ALPHA
2691 007706 100015 BPL LT30C ;AWAIT ALPHA
2692 007710 005300 DEC R0
2693 007712 001373 BNE LT30B
2694 007714 016704 170702 MOV EMADDR,R4
2695 007720 004767 014640 JSR PC,TTOUT ;PRINT HEADER
2696 007724 012704 027412 MOV #MSG50,R4
2697 007730 004767 014630 JSR PC,TTOUT ;PRINT ALPHA ERROR
2698 007734 000167 014132 JMP SCOPE
2699 007740 000435 BR LT30X
2700 007742 012777 000015 170564 LT30C: MOV #15,@MR ;CLOCK MR
2701 007750 012777 000015 170556 MOV #15,@MR ;CLOCK MR
2702 007756 005000 CLR R0
2703 007760 005300 LT30D: DEC R0
2704 007762 001376 BNE LT30D ;DELAY
2705 007764 032777 010000 170532 BIT #10000,@ER ;SEE IF DTE SET
2706 007772 001006 BNE LT30E ;IF SO: BR
2707 007774 012767 000001 170714 MOV #1,EXFL ;SET EXPT FLG
2708 010002 004767 012464 JSR PC,LTGER0 ;GO PRINT ERROR
2709 010006 000412 BR LT30X
2710 010010 012701 010000 LT30E: MOV #10000,R1 ;SET TEST WORD
2711 010014 017702 170504 MOV @ER,R2 ;GET ERROR REGISTER
2712 010020 042702 020000 BIC #20000,R2 ;MASK OPI
2713 010024 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2714 010026 001402 BEQ LT30X ;IF NOT: BR
2715 010030 004767 012424 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2716 010034 004767 014062 LT30X: JSR PC,ITER ;GO SEE IF ITERATION
2717 010040 004767 012212 JSR PC,EORPA ;GO CLEAR GO BIT
2718 010044 004767 013102 JSR PC,DRVCLR
2719 010050 000167 171756 JMP TSCD2 ;RETURN TO SCHED
2720

```



```

2721
2722                                     ;LOGIC TEST 31: OPERATION INCOMPLETE(OPI)*****
2723
2724 010054 012767 031406 170540 LT31:  MOV    #MSLT31,EMADDR  ;SET TEST HEADER
2725 010062 012767 010070 170622      MOV    #LT31IT,SCOLP  ;SET SCOPE ADDRESS
2726 010070 012767 000005 170510 LT31IT: MOV    #5,ITAMT  ;SET REDUCED ITER COUNT
2727 010076 004767 014104      JSR    PC,INIT1     ;INIT, SELECT DRIVE+SLAVE
2728 010102 005000      CLR    R0
2729 010104 005300      LT31W: DEC    R0
2730 010106 001376      BNE    LT31W       ;AWAIT OPI RESET
2731 010110 052777 000300 170424      BIS    #300,@TC    ;SET FORMAT
2732 010116 012777 000013 170410      MOV    #13,@MR     ;SET WAM 2
2733 010124 005077 170366      CLR    @FC         ;SET FCS
2734 010130 012777 000061 170352      MOV    #61,@C1     ;LOAD WRITE+GO
2735 010136 012701 020000      MOV    #20000,R1   ;SET TEST WORD
2736 010142 012703 000004      MOV    #4,R3
2737 010146 005000      CLR    R0
2738 010150 032777 020000 170346 LT31A: BIT    #20000,@ER  ;SEE IE OPI SET
2739 010156 001015      BNE    LT31B       ;IF SO: BR
2740 010160 005300      DEC    R0
2741 010162 001372      BNE    LT31A       ;DELAY
2742 010164 005303      DEC    R3
2743 010166 001370      BNE    LT31A
2744 010170 012767 034252 170474      MOV    #TMS33A,ERADD ;SET ERROR CODE
2745 010176 012767 000001 170512      MOV    #1,EXFL     ;SET EXPT FLG
2746 010204 004767 012262      JSR    PC,LTGERO   ;GO PRINT ERROR
2747 010210 000464      BR     LT31X
2748 010212 017702 170306      LT31B: MOV    @ER,R2   ;GET ERROR REGISTER
2749 010216 020102      CMP    R1,R2       ;SEE IF UNEXPECTED ERRORS
2750 010220 001403      BEQ    LT31C       ;IF NOT: BR
2751 010222 004767 012232      JSR    PC,LTGER3   ;ELSE PRINT ERROR
2752 010226 000455      BR     LT31X
2753 010230 004767 013752      LT31C: JSR    PC,INIT1 ;GO INIT
2754 010234 005000      CLR    R0
2755 010236 005300      LT31W1: DEC    R0
2756 010240 001376      BNE    LT31W1      ;AWAIT OPI RESET
2757 010242 052777 000300 170272      BIS    #300,@TC    ;SET FORMAT
2758 010250 012777 000015 170256      MOV    #15,@MR     ;SET WRAP 3
2759 010256 012777 000071 170224      MOV    #71,@C1     ;LOAD READ+GO
2760 010264 012701 020000      MOV    #20000,R1   ;SET TEST WORD
2761 010270 012703 000100      MOV    #100,R3
2762 010274 005000      CLR    R0
2763 010276 032777 020000 170220 LT31D: BIT    #20000,@ER  ;SEE IF OPI SET
2764 010304 001020      BNE    LT31E       ;IF SO: BR
2765 010306 005300      DEC    R0
2766 010310 001372      BNE    LT31D       ;DELAY
2767 010312 005303      DEC    R3
2768 010314 001370      BNE    LT31D
2769 010316 012767 010230 170366      MOV    #LT31C,SCOLP ;SET SCOPE ADDRESS
2770 010324 012767 034266 170340      MOV    #TMS33B,ERADD ;SET ERROR CODE
2771 010332 012767 000001 170356      MOV    #1,EXFL     ;SET EXPT FLG
2772 010340 004767 012126      JSR    PC,LTGERO   ;GO PRINT ERROR
2773 010344 000406      BR     LT31X
2774 010346 017702 170152      LT31E: MOV    @ER,R2   ;GET ERROR REGISTER
2775 010352 020102      CMP    R1,R2       ;SEE IF UNEXPECTED ERRORS
2776 010354 001402      BEQ    LT31X       ;IF NOT: BR

```

2777	010356	004767	012076		JSR	PC,LTGER3	:ELSE PRINT ERROR
2778	010362	004767	013534		LT31X: JSR	PC,ITER	:GO SEE IF ITERATIONS
2779	010366	004767	012560		JSR	PC,DRVCLR	
2780	010372	012767	000020	170206	MOV	#20,ITAMT	
2781	010400	000167	171426		JMP	TSCD2	:RETURN TO SCHED

```

2782
2783
2784
2785 010404 012767 031442 170210 LT32: MOV #MSLT32,EMADDR ;SET TEST HEADER
2786 010412 012767 010420 170272 MOV #LT32IT,SCOLP ;SET SCOPE ADDRESS
2787 010420 004767 013562 LT32IT: JSR PC,INIT1 ;INIT, SELECT DRIVE +SLAVE
2788 010424 016700 170234 MOV SLVN,R0 ;GET SLAVE NUMBER
2789 010430 005100 COM R0 ;SET NONEXISTANT SLAVE
2790 010432 042700 177770 BIC #177770,R0 ;MASK SLAVE NUMBER
2791 010436 052700 000300 BIS #300,R0 ;SET FORMAT
2792 010442 010077 170074 MOV R0,@TC ;SELECT ILLEGAL SLAVE
2793 010446 012777 000071 170034 MOV #71,@C1 ;LOAD READ+GO
2794 010454 012701 040000 MOV #40000,R1 ;SET TEST WORD
2795 010460 017702 170040 MOV @ER,R2 ;READ ER
2796 010464 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2797 010466 001011 BNE LT32A ;IF SO: BR
2798 010470 012767 034301 170174 MOV #TMS34,ERADD ;SET ERROR CODE
2799 010476 012767 000001 170212 MOV #1,EXFL ;SET ERROR CODE
2800 010504 004767 011762 JSR PC,LTGER0 ;GO PRINT ERROR
2801 010510 000404 BR LT32X
2802 010512 020102 LT32A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2803 010514 001402 BEQ LT32X ;IF NOT: BR
2804 010516 004767 011736 JSR PC,LTGER3 ;ELSE PRINT ERROR
2805 010522 004767 013374 LT32X: JSR PC,ITER ;GO SEE IF ITERATIONS
2806 010526 004767 012420 JSR PC,DRVCLR
2807 010532 000167 171274 JMP TSCD2 ;RETURN TO SCHED
  
```



2808  
2809  
2810  
2811  
2812  
2813  
2814  
2815  
2816  
2817  
2818  
2819  
2820  
2821  
2822  
2823  
2824  
2825  
2826  
2827  
2828

010536 012767 031476 170056  
010544 012767 010552 170140  
010552 004767 013430  
010556 012777 000013 167750  
010564 012777 177777 167724  
010572 012777 000031 167710  
010600 032777 020000 167714  
010606 001010  
010610 012767 034061 170054  
010616 012767 000001 170072  
010624 004767 011642  
010630 004767 013266  
010634 000167 171172

```

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

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

LT33:  MOV      #MSLT33,EMADDR  ;SET TEST HEADER
      MOV      #LT33IT,SCOLP   ;SET SCOPE ADDRESS
LT33IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE+SLAVE
      MOV      #13,@MR         ;SET WAM 2
      MOV      #-1,@FC         ;SET FCS
      MOV      #31,@C1         ;LOAD SPACE FORWARD+GO
      BIT      #20000,@DS      ;SEE IF PIP=1
      BNE     LT33X            ;IF SO: BR
      MOV      #TMS14,ERADD    ;SET ERROR CODE
      MOV      #1,EXFL         ;SET ERROR CODE
      JSR     PC,LTGERO        ;GO PRINT ERROR
LT33X: JSR     PC,ITER         ;GO SEE IF ITERATIONS
      JMP     TSCD2           ;RETURN TO SCHED
```

```

2829
2830
2831
2832 010640 005767 170016      LT34:  TST      NRZOF      ;SEE IF NRZ ONLY
2833 010644 001054              BNE      LT34XX      ;IF SO: BR
2834 010646 012767 034001 170016  MOV      #TMS6,ERADD  ;SET ERROR CODE
2835 010654 012767 031532 167740  MOV      #MSLT34,EMADDR ;SET TEST HEADER
2836 010662 012700 000004      LT34IT: MOV      #4,RO
2837 010666 004767 013314      LT34A1: JSR     PC,INIT1 ;GO INIT, SELECT DRIVE+SLAVE
2838 010672 042777 003400 167642  BIC      #3400,@TC    ;SELECT NRZI
2839 010700 032777 000040 167614  LT34A:  BIT      #40,@DS ;SEE IF PES=0
2840 010706 001410              BEQ      LT34B      ;IF SO: BR
2841 010710 012767 000002 170000  MOV      #2,EXFL     ;SET RCVD-NOT EXPT
2842 010716 012767 010666 167766  MOV      #LT34A1,SCOLP ;SET SCOPE ADDRESS
2843 010724 004767 011542              JSR     PC,LTGERO   ;GO PRINT ERROR
2844 010730 062777 000400 167604  LT34B:  ADD      #400,@TC ;BUMP DENSITY
2845 010736 005300              DEC      RO         ;SEE IF DONE ALL NRZI
2846 010740 001357              BNE      LT34A      ;IF NOT: BR
2847 010742 032777 000040 167552  LT34C:  BIT      #40,@DS ;SEE IF PES=1
2848 010750 001010              BNE      LT34X      ;IF SO: BR
2849 010752 012767 010742 167732  MOV      #LT34C,SCOLP ;SET SCOPE ADDRESS
2850 010760 012767 000001 167730  MOV      #1,EXFL     ;SET EXPT-NOT RCVD FLAG
2851 010766 004767 011500              JSR     PC,LTGERO   ;GO PRINT ERROR
2852 010772 004767 013124      LT34X:  JSR     PC,ITER ;GO SEE IF ITERATION
2853 010776 000167 171030      LT34XX: JMP     TSCD2   ;RETURN TO SCHED

```

```

2854
2855
2856
2857 011002 012767 034324 167662 LT35:  MOV #TMS37,ERADD
2858 011010 012767 031566 167604      MOV #MSLT35,EMADDR
2859 011016 004767 013164      LT35IT: JSR PC,INIT1      ;INIT SELECT DRIVE, SLAVE
2860 011022 032777 000020 167472 1$:  BIT #20,ADS      ;SEE IF SDWN IS RESET
2861 011030 001374      BNE 1$          ;IF NOT: BR
2862 011032 052777 000300 167502      BIS #300,@TC      ;SET FORMAT
2863 011040 012777 000015 167466      MOV #15,@MR       ;SET WAM 3
2864 011046 012777 000071 167434      MOV #71,@C1       ;LOAD READ+GO
2865 011054 032777 020000 167460      BIT #20000,@TC    ;SEE IF TCW=0
2866 011062 001410      BEQ LT35A       ;IF SO: BR
2867 011064 012767 000002 167624      MOV #2,EXFL       ;SET RCV-NOT EXPT FLAG
2868 011072 012767 011016 167612      MOV #LT35IT,SCOLP ;SET SCOPE ADDRESS
2869 011100 004767 011366      JSR PC,LTGERO     ;GO PRINT ERROR
2870 011104 004767 013076      LT35A: JSR PC,INIT1  ;INIT
2871 011110 005077 167426      CLR @TC           ;WRITE TO TC
2872 011114 032777 020000 167420      BIT #20000,@TC    ;SEE IF TCW=1
2873 011122 001010      BNE LT35X        ;IF SO: BR
2874 011124 012767 011104 167560      MOV #LT35A,SCOLP ;SET SCOPE ADDRESS
2875 011132 012767 000001 167556      MOV #1,EXFL       ;SE EXPT-NOT RCVD FLAG
2876 011140 004767 011326      JSR PC,LTGERO     ;GO PRINT ERROR
2877 011144 004767 012752      LT35X: JSR PC,ITER  ;RETURN TO SCHED
2878 011150 000167 170656      JMP TSCD2
    
```



```

2879
2880           ;LOGIC TEST 36: FRAME COUNTER STATUS(FCS)*****
2881
2882 011154 012767 031630 167440 LT36:  MOV    #MSLT36,EMADDR
2883 011162 012767 034332 167502      MOV    #TMS38,ERADD      ;SET ERROR CODE
2884 011170 004767 013012          LT36IT: JSR    PC,INIT1      ;INIT, SELECT DRIVE+SLAVE
2885 011174 032777 040000 167340      BIT    #40000,@TC       ;SEE IF FCS=0
2886 011202 001410          BEQ    LT36A             ;IF SO: BR
2887 011204 012767 011170 167500      MOV    #LT36IT,SCOLP    ;SET SCOPE ADDRESS
2888 011212 012767 000002 167476      MOV    #2,EXFL          ;SET RCVD-NOT EXPT
2889 011220 004767 011246          JSR    PC,LTGERO        ;GO PRINT ERROR
2890 011224 004767 012756          LT36A: JSR    PC,INIT1      ;INIT
2891 011230 005077 167262          CLR    @FC              ;WRITE TO FC
2892 011234 032777 040000 167300      BIT    #40000,@TC       ;SEE IF FCS=1
2893 011242 001010          BNE    LT36X            ;IF SO: BR
2894 011244 012767 011224 167440      MOV    #LT36A,SCOLP    ;SET SCOPE ADDRESS
2895 011252 012767 000001 167436      MOV    #1,EXFL          ;SET EXPT-NOT RCVD
2896 011260 004767 011206          JSR    PC,LTGERO        ;GO PRINT ERROR
2897 011264 004767 012632          LT36X: JSR    PC,ITER      ;GO PRINT ERROR
2898 011270 000167 170536          JMP    TSCD2            ;RETURN TO SCHED

```

```

2899
2900           ;LOGIC TEST 37: ACCELERATION(ACCL)*****
2901
2902 011274 012767 031672 167320 LT37:  MOV   #MSLT37,EMADDR
2903 011302 012767 034340 167362      MOV   #TMS39,ERADD      ;SET ERROR CODE
2904 011310 004767 012672          LT37IT: JSR  PC,INIT1      ;INIT, SELECT DRIVE+SLAVE
2905 011314 052777 000300 167220      BIS   #300,@TC         ;SET FORMAT
2906 011322 005777 167214          TST   @TC              ;SEE IF ACCL=1
2907 011326 100410          BMI   LT37A           ;IF SO: BR
2908 011330 012767 000001 167360      MOV   #1,EXFL
2909 011336 012767 011310 167346      MOV   #LT37IT,SCOLP    ;SET SCOPE ADDRESS
2910 011344 004767 011122          JSR  PC,LTGERO         ;GO PRINT ERROR
2911 011350 004767 012632          LT37A: JSR  PC,INIT1      ;INIT
2912 011354 052777 000300 167160      BIS   #300,@TC         ;SET FORMAT
2913 011362 012777 000015 167144      MOV   #15,@MR          ;SET WAM 3
2914 011370 012777 000071 167112      MOV   #71,@C1          ;LOAD READ+GO
2915 011376 012700 100000          MOV   #100000,R0       ;SET ACCL DELAY
2916 011402 005777 167134          LT37B: TST   @TC         ;SEE IF ACCL=0
2917 011406 100012          BPL   LT37X           ;IF SO: BR
2918 011410 005300          DEC   R0
2919 011412 001373          BNE   LT37B           ;DELAY
2920 011414 012767 011350 167270      MOV   #LT37A,SCOLP    ;SET SCOPE ADDRESS
2921 011422 012767 000002 167266      MOV   #2,EXFL
2922 011430 004767 011036          JSR  PC,LTGERO         ;GO PRINT ERROR
2923 011434 004767 012462          LT37X: JSR  PC,ITER
2924 011440 000167 170366          JMP   TSCD2           ;RETURN TO SCHED
  
```

```

2925
2926
2927
2928 011444 005767 167212          LT40:  TST      NRZOF          ;SEE IF NRZ ONLY
2929 011450 001046                    BNE      LT40XX          ;IF SO: BR
2930 011452 012767 011466 167232    MOV      #LT40IT,SCOLP   ;SET SCOPE ADDRESS
2931 011460 012767 031735 167134    MOV      #MSLT40,EMADDR
2932 011466 004767 012514          LT40IT: JSR      PC,INIT1   ;INIT, SELECT DRIVE+SLAVE
2933 011472 005000                    CLR      R0
2934 011474 005300          LT40W:  DEC      R0
2935 011476 001376                    BNE      LT40W          ;DELAY FOR OPI RESET
2936 011500 052777 002300 167034    BIS      #2300,@TC
2937 011506 012777 000007 167020    MOV      #7,@MR         ;SET WAM 0
2938 011514 012777 000027 166766    MOV      #27,@C1        ;LOAD WRITE TAPE MARK+GO
2939 011522 012700 100000          MOV      #100000,R0     ;SET DELAY
2940 011526 032777 000004 166766    LT40A:  BIT      #4,@DS   ;SEE IF TM=1
2941 011534 001012                    BNE      LT40X          ;IF SO: BR
2942 011536 005300                    DEC      R0
2943 011540 001372                    BNE      LT40A          ;DELAY
2944 011542 012767 033757 167122    MOV      #TMS3,ERADD
2945 011550 012767 000001 167140    MOV      #1,EXFL
2946 011556 004767 010710          JSR      PC,LTGERO      ;GO PRINT ERROR
2947 011562 004767 012334          LT40X:  JSR      PC,ITER
2948 011566 000167 170240          LT40XX: JMP      TSCD2     ;RETURN TO SCHED
  
```



```

2949
2950           ;LOGIC TEST 41: NRZ TAPE MARK (TM,VPE,ITM)*****
2951
2952 011572 012767 011606 167112 LT41:  MOV    #LT41IT,SCOLP  ;SET SCOPE ADDRESS
2953 011600 012767 032002 167014      MOV    #MSLT41,EMADDR
2954 011606 004767 012374      LT41IT: JSR    PC,INIT1    ;INIT, SELECT DRIVE,SLAVE
2955 011612 004767 005436      JSR    PC,BOTT        ;GO ASSURE NOT AT BOT
2956 011616 052777 001700 166716      BIS    #1700,@TC      ;SET NRZ+NORMAL FORMAT
2957 011624 012777 177760 166664      MOV    #-20,@FC       ;SET FCS
2958 011632 012777 000007 166674      MOV    #7,@MR         ;SET WAM 0
2959 011640 012777 000027 166642      MOV    #27,@C1        ;LOAD WRITE TAPE MARK+GO
2960 011646 005000
2961 011650 032777 000004 166644 LT41A:  BIT    #4,@DS        ;SEE IF TM=1
2962 011656 001012      BNE    LT41B          ;IF SO: BR
2963 011660 005300      DEC    R0
2964 011662 001372      BNE    LT41A          ;DELAY
2965 011664 012767 033757 167000      MOV    #TMS3,ERADD    ;SET ERROR CODE
2966 011672 012767 000001 167016      MOV    #1,EXFL
2967 011700 004767 010566      JSR    PC,LTGERO      ;GO PRINT ERROR
2968 011704 032777 002000 166612 LT41B:  BIT    #2000,@ER      ;SEE IF ITM=1
2969 011712 001010      BNE    LT41C          ;IF SO: BR
2970 011714 012767 034222 166750      MOV    #TMS30,ERADD   ;SET ERROR CODE
2971 011722 012767 000001 166766      MOV    #1,EXFL
2972 011730 004767 010536      JSR    PC,LTGERO      ;GO PRINT ERROR
2973 011734 032777 000100 166562 LT41C:  BIT    #100,@ER       ;SEE IF VPE=1
2974 011742 001011      BNE    LT41D          ;IF SO: BR
2975 011744 012767 034207 166720      MOV    #TMS28,ERADD   ;SET ERROR CODE
2976 011752 012767 000001 166736      MOV    #1,EXFL
2977 011760 004767 010506      JSR    PC,LTGERO      ;GO PRINT ERROR
2978 011764 000410      BR     LT41X
2979 011766 012701 002100      LT41D:  MOV    #2100,R1       ;SET EXPT ERROR BITS
2980 011772 017702 166526      MOV    @ER,R2         ;GET ERROR REGISTER
2981 011776 020102      CMP    R1,R2          ;SEE IF UNEXPECTED ERRORS
2982 012000 001402      BEQ    LT41X          ;IF NOT: BR
2983 012002 004767 010452      JSR    PC,LTGER3      ;ELSE PRINT ERROR
2984 012006 005002      LT41X:  CLR    R2             ;SET TIMER
2985 012010 032777 000200 166504 1$:    BIT    #200,@DS       ;SEE IF DRY SET
2986 012016 001002      BNE    2$             ;IF SO: BR
2987 012020 005302      DEC    R2             ;AWAIT DRY
2988 012022 001372      BNE    1$             ;DELAY
2989 012024 004767 012072      2$:    JSR    PC,ITER       ;GO SEE IF ITERATIONS
2990 012030 004767 011116      JSR    PC,DRVCLR      ;GO DO DRIVE CLEAR
2991 012034 000167 167772      JMP    TSCD2          ;RETURN TO SCHED

```

2992  
2993  
2994  
2995  
2996  
2997  
2998  
2999  
3000  
3001  
3002  
3003  
3004  
3005  
3006  
3007  
3008  
3009  
3010  
3011  
3012  
3013  
3014  
3015

012040 000240  
012042 012767 004270 166750  
012050 012767 000100 166744  
012056 012767 010600 166740  
012064 012767 000000 166734  
012072 012767 032051 166522  
012100 012767 001700 166670  
012106 005067 166670  
012112 012767 012120 166572  
012120 000240  
012122 004767 004104  
012126 000240  
012130 005267 166646  
012134 032767 000004 166640  
012142 001766  
012144 000240  
012146 004767 011750  
012152 005067 166630  
012156 000167 167650

LT42:  
LT42A:  
LT42B:

;THE FOLLOWING 13 TESTS WILL CHECK DATA FORMATTING  
;AND TRANSFER THROUGH THE TMO2 WRAP AROUND MODES  
;LOGIC TEST 42: WRAP 3, NRZ, NORMAL ODD \*\*\*\*\*  
NOP  
MOV #4270,WCS1 ;SET EXPT CS1  
MOV #100,WCS2 ;SET EXPT CS2  
MOV #10600,WDS ;SET EXPT DS  
MOV #0,WER ;SET EXPT ER  
MOV #MSLT42,EMADDR ;SET HEADER  
MOV #1700,UDES ;SET NRZ,NORMAL, ODD  
CLR PATRN ;POINT TO PATTERN 0  
MOV #LT42B,SCOLP ;SET SCOPE ADDRESS  
NOP  
JSR PC,WAM3 ;GO DO WRAP 3  
NOP  
INC PATRN ;BUMP PATTERN POINTER  
BIT #4,PATRN ;SEE IF DONE  
BEQ LT42B ;IF NOT: BR  
NOP  
JSR PC,ITER ;GO SEE IF ITERATIONS  
CLR RDRVF ;CLEAR REVENUE FLAG  
JMP TSCD2 ;RETURN TO SCHEDULAR

```
3016
3017
3018
3019 012162 000240
3020 012164 005767 166472
3021 012170 001402
3022 012172 000167 167634
3023 012176 012767 004270 166614
3024 012204 012767 000100 166610
3025 012212 012767 010640 166604
3026 012220 012767 000000 166600
3027 012226 012767 032120 166366
3028 012234 012767 002300 166534
3029 012242 000240
3030 012244 000240
3031 012246 000167 177634

;LOGIC TEST 43: WRAP 3, PE, NORMAL, ODD*****
LT43: NOP
TST NRZOF ;SEE IF NRZ ONLY
BEQ LT43A ;IF NOT: BR
JMP TSCD2 ;RETURN TO SCHED
LT43A: MOV #4270,WCS1 ;SET EXPT CS1
MOV #100,WCS2 ;SET EXPT CS2
MOV #10640,WDS ;SET EXPT DS
MOV #0,WER ;SET EXPT WER
MOV #MSLT43,EMADDR ;SET HEADER
MOV #2300,UDES ;SET PE, NORMAL, ODD
NOP
NOP
JMP LT42A ;EXECUTE TEST SEQUENCE
```



```
3032
3033
3034
3035 012252 000240
3036 012254 012767 004260 166536
3037 012262 012767 000100 166532
3038 012270 012767 010600 166526
3039 012276 012767 000000 166522
3040 012304 012767 032166 166310
3041 012312 012767 001700 166456
3042 012320 005067 166456
3043 012324 012767 012332 166360
3044 012332 000240
3045 012334 004767 003626
3046 012340 000240
3047 012342 005267 166434
3048 012346 032767 000004 166426
3049 012354 001766
3050 012356 000240
3051 012360 004767 011536
3052 012364 000167 167442

;LOGIC TEST 44: WRAP 2, NRZ, NORMAL, ODD*****
LT44: NOP
      MOV #4260,WCS1 ;SET EXPT CS1
      MOV #100,WCS2 ;SET EXPT CS2
      MOV #10600,WDS ;SET EXPT DS
      MOV #0,WER ;SET EXPT WER
      MOV #MSLT44,EMADDR ;SET HEADER
      MOV #1700,UDES ;SET TO NRZ,NORMAL, ODD
LT44A: CLR PATRN ;POINT TO PATTERN 0
      MOV #LT44B,SCOLP ;SET SCOPE ADDRESS
LT44B: NOP
      JSR PC,WAM2 ;GO DO WRAP 2
      NOP
      INC PATRN ;BUMP POINTER
      BIT #4,PATRN ;SEE IF DONE
      BEQ LT44B ;IF NOT: BR
      NOP
      JSR PC,ITER ;GO SEE IF ITERATIONS
      JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
3053
3054
3055
3056 012370 000240
3057 012372 005767 166264
3058 012376 001402
3059 012400 000167 167426
3060 012404 012767 004260 166406
3061 012412 012767 000100 166402
3062 012420 012767 010640 166376
3063 012426 012767 000000 166372
3064 012434 012767 032235 166160
3065 012442 012767 002300 166326
3066 012450 000240
3067 012452 000240
3068 012454 000167 177640

;LOGIC TEST 45: WRAP 2, PE, NORMAL, ODD*****
LT45: NOP
TST NRZOF ;SEE IF NRZ ONLY
BEQ LT45A ;IF NOT: BR
JMP TSCD2 ;RETURN TO SCHED
LT45A: MOV #4260,WCS1 ;SET EXPT CS1
MOV #100,WCS2 ;SET EXPT CS2
MOV #10640,WDS ;SET EXPT DS
MOV #0,WER ;SET EXPT WER
MOV #MSLT45,EMADDR ;SET HEADER
MOV #2300,UDES ;SET PE, NORMAL, ODD
NOP
NOP
JMP LT44A ;GO EXECUTE TEST SEQUENCES
```

```
3069
3070 ;LOGIC TEST 46: WRAP 1, NRZ, NORMAL, ODD*****
3071
3072 012460 000240 LT46: NOP
3073 012462 012767 004260 166330 MOV #4260,WCS1 ;SET EXPT CS1
3074 012470 012767 000100 166324 MOV #100,WCS2 ;SET EXPT CS2
3075 012476 012767 010600 166320 MOV #10600,WDS ;SET EXPT DS
3076 012504 012767 000000 166314 MOV #0,WER ;SET EXPT WER
3077 012512 012767 032303 166102 MOV #MSLT46,EMADDR ;SET HEADER
3078 012520 012767 001700 166250 MOV #1700,UDES ;SET NRZ, NORMAL, ODD
3079 012526 005067 166250 LT46A: CLR PATRN ;POINT TO PATTERN ZERO
3080 012532 012767 012540 166152 MOV #LT46B,SCOLP ;SET SCOPE ADDRESS
3081 012540 000240 LT46B: NOP
3082 012542 004767 003406 JSR PC,WAM1 ;GO DO WRAP 1
3083 012546 000240 NOP
3084 012550 000240 NOP
3085 012552 005267 166224 INC PATRN ;BUMP POINTER
3086 012556 032767 000004 166216 BIT #4,PATRN ;SEE IF DONE
3087 012564 001765 BEQ LT46B ;IF NOT: BR
3088 012566 000240 NOP
3089 012570 004767 011326 JSR PC,ITER ;GO SEE IF ITERATIONS
3090 012574 000167 167232 JMP TSCD2 ;RETURN TO SCHEDULAR
```



```
3091
3092
3093
3094 012600 000240
3095 012602 005767 166054
3096 012606 001402
3097 012610 000167 167216
3098 012614 004767 007364
3099 012620 012767 004260 166172
3100 012626 012767 000100 166166
3101 012634 012767 010640 166162
3102 012642 012767 000000 166156
3103 012650 012767 032352 165744
3104 012656 012767 002300 166112
3105 012664 000240
3106 012666 000240
3107 012670 000167 177632

;LOGIC TEST 47: WRAP 1, PE, NORMAL, ODD*****
LT47: NOP
      TST NRZOF ;SEE IF NRZ ONLY
      BEQ LT47A ;IF NOT: BR
      JMP TSCD2 ;RETURN TO SCHED
LT47A: JSR PC,PPGEN ;GO GENERATE PRE/POSTAMBLE
      MOV #4260,WCS1 ;SET EXPT CS1
      MOV #100,WCS2 ;SET EXPT CS2
      MOV #10640,WDS ;SET EXPT DS
      MOV #0,WER ;SET EXPT WER
      MOV #MSLT47,EMADDR ;SET HEADER
      MOV #2300,UDES ;SET PE, NORMAL, ODD
      NOP
      NOP
      JMP LT46A ;GO EXECUTE TEST SEQUENCE
```

```
3108
3109
3110 ;LOGIC TEST 50: WRAP 0, NRZ,NORMAL, ODD*****
3111 012674 000240 LT50: NOP
3112 012676 012767 144260 166114 MOV #144260,WCS1 ;SET EXPT CS1
3113 012704 012767 000100 166110 MOV #100,WCS2 ;SET EXPT CS2
3114 012712 012767 150600 166104 MOV #150600,WDS ;SET EXPT DS
3115 012720 012767 000200 166100 MOV #200,WER ;SET EXPT ER
3116 012726 012767 032420 165666 MOV #MSLT50,EMADDR ;SET HEADER
3117 012734 012767 001700 166034 MOV #1700,UDES ;SET NRZ, NORMAL, ODD
3118 012742 005067 166034 LT50A: CLR PATRN ;POINT TO PATTERN 0
3119 012746 012767 012754 165736 MOV #LT50B,SCOLP ;SET SCOPE ADDRESS
3120 012754 000240 LT50B: NOP
3121 012756 004767 003126 JSR PC,WAMO ;GO DO WRAP 0
3122 012762 000240 NOP
3123 012764 005267 166012 INC PATRN ;BUMP POINTER
3124 012770 032767 000004 166004 BIT #4,PATRN ;SEE IF DONE
3125 012776 001766 BEQ LT50B ;IF NOT: BR
3126 013000 000240 NOP
3127 013002 004767 011114 JSR PC,ITER ;GO SEE IF ITERATIONS
3128 013006 000167 167020 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
3129
3130
3131
3132 013012 000240
3133 013014 005767 165642
3134 013020 001402
3135 013022 000167 167004
3136 013026 012767 004260 165764
3137 013034 012767 000100 165760
3138 013042 012767 010640 165754
3139 013050 012767 000000 165750
3140 013056 012767 032467 165536
3141 013064 012767 002300 165704
3142 013072 000240
3143 013074 000240
3144 013076 000167 177640

;LOGIC TEST 51: WRAP 0, PE, NORMAL, ODD*****
LT51: NOP
      TST NRZOF ;SEE IF NRZ ONLY
      BEQ LT51A ;IF NOT: BR
      JMP TSCD2 ;RETURN TI SCHED
LT51A: MOV #4260,WCS1 ;SET EXPT CS1
      MOV #100,WCS2 ;SET EXPT CS2
      MOV #10640,WDS ;SET EXPT DS
      MOV #0,WER ;SET EXPT ER
      MOV #MSLT51,EMADDR ;SET HEADER
      MOV #2300,UDES ;SET PE, NORMAL, ODD
      NOP
      NOP
      JMP LT50A ;GO EXECUTE TEST SEQUENCE
```



```
3145
3146 ;LOGIC TEST 52: CORE DUMP WRITE, WAM2*****
3147
3148 013102 000240 LT52: NOP
3149 013104 012767 004260 165706 MOV #4260,WCS1 ;SET EXPT CS1
3150 013112 012767 000100 165702 MOV #100,WCS2 ;SET EXPT CS2
3151 013120 012767 010600 165676 MOV #10600,WDS ;SET EXPT DS
3152 013126 012767 000000 165672 MOV #0,WER ;SET EXPT ER
3153 013134 012767 032535 165460 MOV #MSLT52,EMADDR ;SET HEADER
3154 013142 012767 001720 165626 MOV #1720,UDES ;SET NRZ, CORE DUMP, ODD
3155 013150 005067 165626 CLR PATRN ;POINT TO PATTERN 0
3156 013154 012767 013162 165530 MOV #LT52A,SCOLP ;SET SCOPE ADDRESS
3157 013162 000240 LT52A: NOP
3158 013164 004767 002776 JSR PC,WAM2 ;GO DO WAM 2
3159 013170 000240 NOP
3160 013172 022767 000002 165602 CMP #2,PATRN ;SEE IF DONE
3161 013200 001404 BEQ LT52X ;IF SO: BR
3162 013202 012767 000002 165572 MOV #2,PATRN ;SELECT PATTERN 2
3163 013210 000764 BR LT52A ;CONTINUE
3164 013212 000240 LT52X: NOP
3165 013214 004767 010702 JSR PC,ITER ;GO SEE IF ITERATIONS
3166 013220 000167 166606 JMP TSCD2 ;RETURN TO SCHEDULES
3167
```

```
3168
3169 ;LOGIC TEST 53: CORE DUMP READ, WAM 3*****
3170
3171 013224 000240 LT53: NOP
3172 013226 012767 004270 165564 MOV #4270,WCS1 ;SET EXPT CS1
3173 013234 012767 000100 165560 MOV #100,WCS2 ;SET EXPT CS2
3174 013242 012767 010600 165554 MOV #10600,WDS ;SET EXPT DS
3175 013250 012767 000000 165550 MOV #0,WER ;SET EXPT ER
3176 013256 012767 032606 165336 MOV #MSLT53,EMADDR ;SET HEADER
3177 013264 012767 001720 165504 MOV #1720,UDES ;SELECT NRE, CORE DUMP, ODD
3178 013272 005067 165504 CLR PATRN ;SELECT PATTERN 0
3179 013276 012767 013312 165406 MOV #LT53A,SCOLP ;SET SCOPE ADDRESS
3180 013304 012767 001054 165476 MOV #WCDP0,RCDP ;POINT TO PATTERN 0
3181 013312 000240 LT53A: NOP
3182 013314 004767 002712 JSR PC,WAM3 ;GO DO WAM3
3183 013320 000240 NOP
3184 013322 022767 000002 165452 CMP #2,PATRN ;SEE IF DONE
3185 013330 001407 BEQ LT53X ;IF SO: BR
3186 013332 012767 000002 165442 MOV #2,PATRN ;SELECT PATTERN 2
3187 013340 012767 001042 165442 MOV #WCDP2,RCDP ;POINT TO PATTERN 2
3188 013346 000761 BR LT53A ;CONTINUE
3189 013350 004767 010546 LT53X: JSR PC,ITER ;GO SEE IF ITERATION
3190 013354 000167 166452 JMP TSCD2 ;RETURN TO SCHEDULE
```

```
3191  
3192  
3193  
3194 013360 000240  
3195 013362 012767 004260 165430  
3196 013370 012767 000100 165424  
3197 013376 012767 010600 165420  
3198 013404 012767 000000 165414  
3199 013412 012767 032656 165202  
3200 013420 012767 001710 165350  
3201 013426 000240  
3202 013430 000240  
3203 013432 000167 177070
```

;LOGIC TEST 54: EVEN PARITY WRITE: WAM 1(M8903)\*\*\*\*\*

```
LT54:  NOP  
      MOV #4260,WCS1      ;SET EXPT CS1  
      MOV #100,WCS2      ;SET EXPT CS2  
      MOV #10600,WDS     ;SET EXPT DS  
      MOV #0,WER         ;SET EXPT ER  
      MOV #MSLT54,EMADDR ;SET HEADER  
      MOV #1710,UDES     ;SET NRZ, NORMAL, EVEN  
      NOP  
      NOP  
      JMP LT46A          ;GO EXECUTE WAM 1
```



```
3204  
3205  
3206  
3207 013436 000240  
3208 013440 012767 144260 165352  
3209 013446 012767 000100 165346  
3210 013454 012767 150600 165342  
3211 013462 012767 000200 165336  
3212 013470 012767 032737 165124  
3213 013476 012767 001710 165272  
3214 013504 000240  
3215 013506 000240  
3216 013510 000167 177226
```

;LOGIC TEST 55: EVEN PARITY READ: WAM 0(M8903 M8904)\*\*\*\*\*

```
LT55: NOP  
MOV #144260,WCS1 :SET EXPT CS1  
MOV #100,WCS2 :SET EXPT CS2  
MOV #150600,WDS :SET EXPT DS  
MOV #200,WER :SET EXPT ER  
MOV #MSLT55,EMADDR :SET HEADER  
MOV #1710,UDES :SET NRZ, NORMAL, EVEN  
NOP  
NOP  
JMP LT50A :GO DO WAM 0
```

```
3217
3218
3219
3220 013514 000240          LT56:  NOP
3221 013516 012767 004276 165274  MOV   #4276,WCS1      :SET EXPT CS1
3222 013524 012767 000100 165270  MOV   #100,WCS2       :SET EXPT CS2
3223 013532 012767 010640 165264  MOV   #10640,WDS      :SET EXPT DS
3224 013540 012767 000000 165260  MOV   #0,WER           :SET EXPT ER
3225 013546 012767 033016 165046  MOV   #MSLT56,EMADDR  :SET HEADER
3226 013554 012767 002300 165214  MOV   #2300,UDES      :SELECT PE,NORMAL,ODD
3227 013562 012767 000001 165216  MOV   #1,RDRVF        :SET READ REVERSE FLAG
3228 013570 000240          NOP
3229 013572 000240          NOP
3230 013574 000167 176306          JMP   LT42A           :GO DO WAM 3, REVERSE
3231
```

3232  
3233  
3234  
3235  
3236  
3237  
3238  
3239 013600 000240  
3240 013602 004767 002006  
3241 013606 012700 001000  
3242 013612 005300  
3243 013614 001376  
3244 013616 012767 033063 164776  
3245 013624 012767 013632 165060  
3246 013632 004767 010350  
3247 013636 052777 001700 164676  
3248 013644 012777 177770 164640  
3249 013652 012777 177760 164636  
3250 013660 012777 034350 164626  
3251 013666 012777 000007 164640  
3252 013674 012777 000061 164606  
3253 013702 005000  
3254 013704 032777 000200 164610  
3255 013712 001002  
3256 013714 005300  
3257 013716 001372  
3258 013720 000240  
3259 013722 022777 000200 164574  
3260 013730 001007  
3261 013732 017702 164572  
3262 013736 042702 177000  
3263 013742 022702 000777  
3264 013746 001410  
3265 013750 004767 006504  
3266 013754 012704 027517  
3267 013760 004767 010600  
3268 013764 000167 166042  
3269 013770 004767 010212  
3270 013774 052777 000300 164540  
3271 014002 012777 177770 164502  
3272 014010 012777 177760 164500  
3273 014016 012777 034350 164470  
3274 014024 012777 000021 164502  
3275 014032 012777 000061 164450  
3276 014040 005000  
3277 014042 032777 000200 164452  
3278 014050 001002  
3279 014052 005300  
3280 014054 001372  
3281 014056 000240  
3282 014060 005777 164440  
3283 014064 100411  
3284 014066 012767 034316 164576  
3285 014074 012767 000001 164614  
3286 014102 004767 006364  
3287 014106 000410

:THE FOLLOWING SIX(6) TEST WILL REQUIRE TAPE MOVEMENT. EACH  
:TEST WILL PERFORM A TAPE WRITE WHILE IN A PARTICULAR MAINTENANCE  
:MODE IN ORDER TO FORCE THE REMAINING ERROR CONDITIONS.

:LOGIC TEST 57: CYCLIC REDUNDANCY ERROR(CRC)\*\*\*\*\*

LT57: NOP  
JSR PC,STATIC ;GO SEE IF STATIC ONLY  
MOV #1000,R0  
LT57PS: DEC R0  
BNE LT57PS ;PAUSE  
MOV #MSLT57,EMADDR  
MOV #LT57IT,SCOLP ;SET SCOPE ADDRESS  
LT57IT: JSR PC,INIT1 ;INIT SELECT DRIVE+SLAVE  
BIS #1700,@TC ;SET NRZ + NORMAL FORMAT  
MOV #-10,@WC  
MOV #-20,@FC ;SET FC=20  
MOV #WDATA,@BA ;SET BUS ADDRESS  
MOV #7,@MR ;SET MM CODE  
MOV #61,@C1 ;LOAD WRITE+GO  
CLR R0  
LT57A: BIT #200,@DS ;SEE IF DRY=1  
BNE LT57B ;IF SO: BR  
DEC R0  
BNE LT57A ;DELAY  
LT57B: NOP  
CMP #200,@ER ;SEE IF LRC ERROR ONLY  
BNE LT57B1 ;IF NOT: BR  
MOV @CC,R2 ;GET CHECK CHAR  
BIC #177000,R2 ;MASK CRC  
CMP #777,R2 ;SEE IF SETUP CRC IS CORRECT  
BEQ LT57B2 ;IF SO: BR  
LT57B1: JSR PC,LTGER3 ;ELSE PRINT ERROR SETUP  
MOV #MSG55,R4  
JSR PC,TTOUT ;PRINT SETUP ERROR MSG  
JMP TSCD2 ;RETURN TO SCHED  
LT57B2: JSR PC,INIT1 ;GO INIT  
BIS #300,@TC ;SET FORMAT+NRZ  
MOV #-10,@WC ;SET WC  
MOV #-20,@FC ;SET FC  
MOV #WDATA,@BA ;SET BA  
MOV #21,@MR ;SET MM  
MOV #61,@C1 ;LOAD WRITE+GO  
CLR R0  
LT57C: BIT #200,@DS ;SEE IF DRY  
BNE LT57D ;IF SO: BR  
DEC R0  
BNE LT57C ;AWAIT DRY  
LT57D: NOP  
TST @ER ;SEE IF CRC=1  
BMI LT57E ;IF SO: BR  
MOV #TMS36,ERADD ;SET ERROR CODE  
MOV #1,EXFL  
JSR PC,LTGER0 ;GO PRINT ERROR  
BR LT57X



3288	014110	012701	100200	LT57E:	MOV	#100200,R1	;SET EXPT ERROR BITS
3289	014114	017702	164404		MOV	@ER,R2	;GET ERROR REGISTER
3290	014120	020102			CMP	R1,R2	;SEE IF UNEXPECTED ERRORS
3291	014122	001402			BEQ	LT57X	;IF NOT: BR
3292	014124	004767	006330		JSR	PC,LTGER3	;ELSE PRINT ERROR
3293	014130	000240		LT57X:	NOP		
3294	014132	004767	007764		JSR	PC,ITER	;DO ITERATIONS
3295	014136	004767	007010		JSR	PC,DRVCLR	
3296	014142	000167	165664		JMP	TSCD2	;RETURN TO SCHED

```

3297
3298
3299
3300 014146 000240          LT60:  NOP
3301 014150 004767 001440  JSR      PC,STATIC          :GO SEE IF STATIC ONLY
3302 014154 012767 014170 164530  MOV     #LT60IT,SCOLP      :SET SCOPE ADDRESS
3303 014162 012767 033117 164432  MOV     #MSLT60,EMADDR
3304 014170 004767 010012          LT60IT: JSR     PC,INIT1        :INIT, SELECT DRIVE+SLAVE
3305 014174 052777 000300 164340  BIS     #300,@TC           :SET FORMAT+NRZ
3306 014202 012777 000023 164324  MOV     #23,@MR            :SET MM
3307 014210 012777 177770 164274  MOV     #-10,@WC           :SET WC
3308 014216 012777 177760 164272  MOV     #-20,@FC           :SET FC
3309 014224 012777 034350 164262  MOV     #WDATA,@BA        :SET BA
3310 014232 012777 000061 164250  MOV     #61,@C1           :LOAD WRITE+GO
3311 014240 005000          CLR     R0
3312 014242 032777 000200 164252  LT60C: BIT     #200,@DS      :SEE IF DRY
3313 014250 001002          BNE     LT60D              :IF SO: BR
3314 014252 005300          DEC     R0
3315 014254 001372          BNE     LT60C              :AWAIT DRY
3316 014256 000240          LT60D: NOP
3317 014260 032777 000200 164236  BIT     #200,@ER           :SEE IF LRC=1
3318 014266 001011          BNE     LT60E              :IF SO: BR
3319 014270 012767 034173 164374  MOV     #TMS26,ERADD      :SET ERROR CODE
3320 014276 012767 000001 164412  MOV     #1,EXFL
3321 014304 004767 006162          JSR     PC,LTGER0         :GO PRINT
3322 014310 000425          BR      LT60X
3323 014312 017702 164216          LT60E: MOV     @MR,R2
3324 014316 042702 000177          BIC     #177,R2           :MASK LRC
3325 014322 012701 157600          MOV     #157600,R1        :SET EXPT LRC
3326 014326 020102          CMP     R1,R2             :SEE IF EXPT = RCVD
3327 014330 001405          BEQ     LT60F              :IF SO: BR
3328 014332 012767 027474 164332  MOV     #MSG53,ERADD      :SET ERROR CODE
3329 014340 004767 007302          JSR     PC,LTGER1         :PRINT ERROR
3330 014344 017702 164154          LT60F: MOV     @ER,R2           :GET ERROR REGISTER
3331 014350 012701 000200          MOV     #200,R1           :SET EXPT ERROR BITS
3332 014354 020102          CMP     R1,R2             :SEE IF UNEXPECTED ERRORS
3333 014356 001402          BEQ     LT60X              :IF NOT: BR
3334 014360 004767 006074          JSR     PC,LTGER3         :ELSE PRINT ERROR
3335 014364 000240          LT60X: NOP
3336 014366 004767 007530          JSR     PC,ITER
3337 014372 004767 006554          JSR     PC,DRVCLR
3338 014376 000167 165430          JMP     TSCD2             :RETURN TO SCHED

```

```

3339
3340 ;LOGIC TEST 61: PE CORRECTABLE DATA (CORR)*****
3341
3342 014402 000240 LT61: NOP
3343 014404 005767 164252 TST NRZOF ;SEE IF NRZ ONLY
3344 014410 001124 BNE LT61XX ;IF SO: BR
3345 014412 004767 001176 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3346 014416 012767 033153 164176 MOV #MSLT61,EMADDR
3347 014424 012767 014432 164260 MOV #LT61IT,SCOLP
3348 014432 004767 007550 LT61IT: JSR PC,INIT1 ;INIT, SELCT DRIVE+SLAVE
3349 014436 052777 002300 164076 BIS #2300,@TC ;SET PE,NORMAL
3350 014444 012777 177600 164040 MOV #-200,@WC ;SET WC=200
3351 014452 012777 177400 164036 MOV #-400,@FC ;SET FC=400
3352 014460 012777 034350 164026 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
3353 014466 012777 000061 164014 MOV #61,@C1 ;LOAD WRITE+GO
3354 014474 005000 CLR RO
3355 014476 005777 164014 LT61A: TST @FC ;SEE IF FC=0
3356 014502 001402 BEQ LT61A1 ;IF SO: BR
3357 014504 005300 DEC RO
3358 014506 001373 BNE LT61A ;DELAY FOR FC=0
3359 014510 012777 000021 164016 LT61A1: MOV #21,@MR ;SET MAINT CODE
3360 014516 005000 CLR RO
3361 014520 032777 000200 163774 LT61B: BIT #200,@DS ;SEE IF DRY IS SET
3362 014526 001002 BNE LT61C ;IF SO: BR
3363 014530 005300 DEC RO
3364 014532 001372 BNE LT61B ;AWAIT DRY
3365 014534 000240 LT61C: NOP
3366 014536 005777 163762 TST @ER ;SEE IF CORR=1
3367 014542 100410 BMI LT61D ;IF SO: BR
3368 014544 012767 034307 164120 MOV #TMS35,ERADD ;SET ERROR CODE
3369 014552 012767 000001 164136 MOV #1,EXFL
3370 014560 004767 005706 JSR PC,LTGERO ;GO PRINT ERROR
3371 014564 000240 LT61D: NOP
3372 014566 000240 LT61E: NOP
3373 014570 122777 000002 163732 CMPB #2,@CC ;SEE IF DEAD TRACK BIT 1
3374 014576 001414 BEQ LT61F ;IF SO: BR
3375 014600 117702 163724 MOVB @CC,R2 ;SAVE RCVD
3376 014604 042702 177000 BIC #177000,R2 ;MASK OUT CRC
3377 014610 112701 000002 MOVB #2,R1 ;SAVE EXPT
3378 014614 012767 027067 164050 MOV #MSG42,ERADD ;SET ERROR CODE
3379 014622 004767 007020 JSR PC,LTGER1 ;GO PRINT ERROR
3380 014626 000410 BR LT61X
3381 014630 017702 163670 LT61F: MOV @ER,R2 ;GET ERROR REGISTER
3382 014634 012701 100000 MOV #100000,R1 ;SET EXPT ERROR BITS
3383 014640 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3384 014642 001402 BEQ LT61X ;IF NOT: BR
3385 014644 004767 005610 JSR PC,LTGER3 ;ELSE PRINT ERROR
3386 014650 000240 LT61X: NOP
3387 014652 004767 007244 JSR PC,ITER
3388 014656 004767 006270 JSR PC,DRVCLR
3389 014662 000167 165144 LT61XX: JMP TSCD2 ;RETURN TO SCHED
3390

```



```

3391                                     ;LOGIC TEST 62: PE INCORRECTABLE DATA(INC)*****
3392
3393 014666 005767 163770                LT62:  TST      NRZOF          ;SEE IF NRZ ONLY
3394 014672 001120                        BNE      LT62XX          ;IF SO: BR
3395 014674 004767 000714                JSR      PC,STATIC      ;GO SEE IF STATIC ONLY
3396 014700 012767 033233 163714          MOV      #MSLT62,EMADDR
3397 014706 012767 014714 163776          MOV      #LT62IT,SCOLP
3398 014714 004767 007266                LT62IT: JSR      PC,INIT1 ;INIT SELECT DRIVE SLAVE
3399 014720 012777 177560 163564          MOV      #-220,@WC      ;SET WC=220
3400 014726 012777 177340 163562          MOV      #-440,@FC      ;SET FC=440
3401 014734 012777 034350 163552          MOV      #WDATA,@BA     ;SET BA=START OF WRITE BUFFER
3402 014742 052777 002300 163572          BIS      #2300,@TC      ;SET TO PE,NORMAL
3403 014750 012777 000061 163532          MOV      #61,@C1        ;LOAD WRITE+GO
3404 014756 005000                        CLR      RO
3405 014760 005777 163532                LT62E:  TST      @FC          ;AWAIT FC=0
3406 014764 001402                        BEQ      LT62E1
3407 014766 005300                        DEC      RO
3408 014770 001373                        BNE      LT62E          ;AWAIT FC=0
3409 014772 012777 000023 163534          LT62E1: MOV      #23,@MR      ;SET MAINT CODE
3410 015000 005000                        CLR      RO
3411 015002 032777 000200 163512          LT62A:  BIT      #200,@DS   ;SEE IF DRY IS SET
3412 015010 001002                        BNE      LT62B          ;IF SO: BR
3413 015012 005300                        DEC      RO
3414 015014 001372                        BNE      LT62A          ;AWAIT DRY
3415 015016 032777 000100 163500          LT62B:  BIT      #100,@ER   ;SEE IF INC=1
3416 015024 001010                        BNE      LT62D          ;IF SO:BR
3417 015026 012767 034151 163636          MOV      #TMS23,ERADD   ;SET ERROR CODE
3418 015034 012767 000001 163654          MOV      #1,EXFL
3419 015042 004767 005424                JSR      PC,LTGER0      ;GO PRINT ERROR
3420 015046 017702 163456                LT62D:  MOV      @CC,R2    ;GET CHECK CHAR
3421 015052 042702 177000                BIC      #177000,R2     ;MASK CHECK CHAR
3422 015056 012701 000046                MOV      #46,R1        ;SET EXPT CK
3423 015062 020102                        CMP      R1,R2          ;SEE IF EXPT = RCVD
3424 015064 001405                        BEQ      LT62F          ;IF SO: BR
3425 015066 012767 027506 163576          MOV      #MSG54,ERADD   ;ELSE GO PRINT ERROR
3426 015074 004767 006546                JSR      PC,LTGER1
3427 015100 017702 163420                LT62F:  MOV      @ER,R2    ;MASK MSG AND CORR AND PEF
3428 015104 042702 100600                BIC      #100600,R2     ;SET EXPT ERROR BITS
3429 015110 012701 000100                MOV      #100,R1       ;SEE IF UNEXPECTED ERRORS
3430 015114 020102                        CMP      R1,R2
3431 015116 001402                        BEQ      LT62X          ;IF NOT: BR
3432 015120 004767 005334                JSR      PC,LTGER3     ;ELSE PRINT ERROR
3433 015124 004767 006772                LT62X:  JSR      PC,ITER
3434 015130 004767 006016                JSR      PC,DRVCLR
3435 015134 000167 164672                LT62XX: JMP      TSCD2    ;RETURN TO SCHED

```

```

3436
3437
3438
3439 015140 000240          LT63:  NOP
3440 015142 005767 163514    TST     NRZOF          ;SEE IF NRZ ONLY
3441 015146 001121          BNE     LT63XX        ;IF SO: BR
3442 015150 004767 000440    JSR     PC,STATIC     ;GO SEE IF STATIC ONLY
3443 015154 012767 033315 163440  MOV     #MSLT63,EMADDR ;SET HEADER
3444 015162 012767 015170 163522  MOV     #LT63IT,SCOLP  ;SET SCOPE ADDRESS
3445 015170 004767 007012          LT63IT: JSR     PC,INIT1    ;INITIALIZE
3446 015174 012777 177770 163310  MOV     #-10,@WC      ;SET WC=10
3447 015202 012777 177760 163306  MOV     #-20,@FC      ;SET FC=20
3448 015210 052777 002300 163324  BIS     #2300,@TC     ;SET TO PE,NORMAL
3449 015216 012777 034350 163270  MOV     #WDATA,@BA    ;SET BA=START OF WRITE BUFFER
3450 015224 012777 000061 163256  MOV     #61,@C1       ;LOAD WRITE+GO
3451 015232 005777 163260          LT63A:  TST     @FC     ;SEE IF FIRST FRAME OUT
3452 015236 001375          BNE     LT63A        ;IF SO: BR
3453 015240 032777 000100 163266  1$:    BIT     #100,@MR     ;
3454 015246 001774          BEQ     1$           ;DELAY
3455 015250 032777 000100 163256  2$:    BIT     #100,@MR     ;
3456 015256 001374          BNE     2$           ;
3457 015260 012777 000027 163246  MOV     #27,@MR       ;SET MM CODE TO KILL PEF
3458 015266 005000          CLR     R0           ;
3459 015270 012767 007640 000120  MOV     #4000.,LT630  ;
3460 015276 032777 000200 163216  LT63B: BIT     #200,@DS  ;SEE IF DRY SET
3461 015304 001005          BNE     LT63C        ;IF SO: BR
3462 015306 005300          DEC     R0           ;
3463 015310 001372          BNE     LT63B        ;AWAIT DRY
3464 015312 005367 000100    DEC     LT630        ;
3465 015316 001367          BNE     LT63B        ;
3466 015320 000240          LT63C:  NOP
3467 015322 032777 000200 163174  BIT     #200,@ER     ;SEE IF PEF SET
3468 015330 001011          BNE     LT63D        ;IF SO: BR
3469 015332 012767 034165 163332  MOV     #TMS25,ERADD  ;SET ERROR TAG
3470 015340 012767 000001 163350  MOV     #1,EXFL       ;SET EXPT FLAG
3471 015346 004767 005120    JSR     PC,LTGERO    ;GO PRINT ERROR
3472 015352 000412          BR      LT63X        ;
3473 015354 017702 163144    LT63D:  MOV     @ER,R2   ;GET ERROR REGISTER
3474 015360 042702 102500    BIC     #102500,R2   ;**MASK NSG
3475 015364 012701 000200    MOV     #200,R1     ;SET EXPT ERROR BITS
3476 015370 020102          CMP     R1,R2       ;SEE IF UNEXPECTED ERRORS
3477 015372 001402          BEQ     LT63X        ;IF NOT: BR
3478 015374 004767 005060    JSR     PC,LTGER3   ;ELSE PRINT ERROR
3479 015400 000240          LT63X:  NOP
3480 015402 004767 006514    JSR     PC,ITER     ;
3481 015406 004767 005540    JSR     PC,DRVCLR   ;
3482 015412 000167 164414    LT63XX: JMP     TSCD2   ;RETURN TO SCHED
3483
3484 015416 000000          LT630:  0

```



```

3485                                     ;LOGIC TEST 64: FRAME COUNT OVERFLOW(M8905)*****
3486
3487 015420 000240                       LT64:  NOP
3488 015422 012767 033351 163172         MOV    #MSLT64,EMADDR  ;SET TEST HEADER
3489 015430 012767 015436 163254         MOV    #LT64IT,SCOLP  ;SET SCOPE ADDRESS
3490 015436 004767 006544                 LT64IT: JSR   PC,INIT1  ;GO INIT
3491 015442 012777 177770 163042         MOV    #-10,@WC       ;SET WC = 10
3492 015450 012777 177760 163040         MOV    #-20,@FC       ;SET FC = 20
3493 015456 052777 001700 163056         BIS    #1700,@TC      ;SET TO NRZ, NORMAL, ODD
3494 015464 012777 034350 163022         MOV    #WDATA,@BA     ;SET BUS ADDRESS
3495 015472 012777 000013 163034         MOV    #13,@MR        ;SET WRAP 2
3496 015500 012777 000061 163002         MOV    #61,@C1        ;LOAD WRITE+GO
3497 015506 012700 040000                 MOV    #40000,R0
3498 015512 005777 163024                 LT64A: TST   @TC       ;SEE IF ALPHA
3499 015516 100002                         BPL    LT64B          ;IF SO: BR
3500 015520 005300                         DEC    R0
3501 015522 001373                         BNE    LT64A          ;AWAIT ALPHA
3502 015524 012700 000020                 LT64B: MOV   #20,R0    ;SET CLK CNT
3503 015530 052777 000040 162776         LT64C: BIS   #40,@MR
3504 015536 042777 000040 162770         BIC   #40,@MR        ;CLOCK MR
3505 015544 005300                         DEC    R0
3506 015546 001370                         BNE    LT64C          ;IF NOT DONE ALL: BR
3507 015550 017702 162742                 MOV   @FC,R2
3508 015554 005001                         CLR   R1              ;SET TEST WORD
3509 015556 020102                         CMP   R1,R2           ;SEE IF EXPT = RCVD
3510 015560 001410                         BEQ   LT64X           ;IF SO: BR
3511 015562 012767 026342 163102         MOV   #MSG19,ERADD   ;SET ERROR CODE
3512 015570 012767 000001 163120         MOV   #1,EXFL        ;SET EXPT FLAG
3513 015576 004767 006044                 JSR   PC,LTGER1      ;GO PRINT ERROR
3514 015602 000240                       LT64X: NOP
3515 015604 004767 006312                 JSR   PC,ITER        ;GO SEE IF ITERATIONS
3516 015610 000167 164216                 JMP   TSCD2          ;RETURN TO SCHEDULAR
3517
3518                                     ;STATIC TESTS ONLY SUBROUTINE*****
3519
3520 015614 005767 163120                 STATIC: TST   STFLG   ;SEE IF SINGLE TEST ONLY
3521 015620 001006                         BNE    STATX        ;IF SO: BR
3522 015622 005767 163164                 TST   STATC         ;SEE IF STATIC ONLY
3523 015626 001403                         BEQ    STATX        ;IF NOT: BR
3524 015630 005726                         TST   (SP)+         ;RESET STACK
3525 015632 000167 164174                 JMP   TSCD2          ;RETURN TO SCHEDULAR
3526 015636 000207                         STATX: RTS   PC      ;RETURN TO TEST
3527

```



```
3528
3529
3530
3531
3532
3533
3534
3535
3536 015640 000240          DSUP:  NOP
3537 015642 005767 163072    TST      STFLG          ;SEE IF SINGLE TEST
3538 015646 001431          BEQ      DSO            ;IF NOT: BR
3539 015650 032777 000100 162712 BIT      #100,@SWR      ;SEE IF SELECT PATTERN
3540 015656 001425          BEQ      DSO            ;IF NOT: BR
3541 015660 012704 034766    MOV      #WMSG3,R4
3542 015664 004767 006674    JSR      PC,TTOUT      ;REQUEST PATTERN NUMBER
3543 015670 016703 163106    MOV      PATRN,R3
3544 015674 004767 007012    JSR      PC,OCTP       ;PRINT PATTERN NUMBER
3545 015700 012705 001002    MOV      #PATRN,R5    ;GET ADDRESS OF PATRN ENTRY
3546 015704 012701 000001    MOV      #1,R1        ;SET SIZE OF ENTRY
3547 015710 012702 000003    MOV      #3,R2        ;SET UPPER LIMIT
3548 015714 012703 000000    MOV      #0,R3        ;SET LOWER LIMIT
3549 015720 004767 006402    JSR      PC,TTR        ;GO GET PATTERN NUMBER
3550 015724 012767 000001 163046 MOV      #1,WPGFL      ;SET FLAG
3551 015732 012703 035646    DS0:    MOV      #WBUF,R3   ;R3 = ADDR OF WRITE BUFFER
3552 015736 016701 163040    MOV      PATRN,R1     ;R1 = PATTERN SELECTOR
3553 015742 062701 000001    ADD      #1,R1        ;BUMP POINTER
3554 015746 000241          CLC
3555 015750 006101          ROL      R1            ;MAKE PATTERN SELECTOR EVEN
3556 015752 000171 001030    JMP      @DATBL(R1)   ;GO GENERATE PATTERN
3557 015756 032777 010000 162552 DS1:    BIT      #10000,@DT   ;SEE IF SEVEN TRACK
3558 015764 001410          BEQ      DS3          ;IF NOT: BR
3559 015766 012702 000202    MOV      #202,R2     ;SET BUFFER SIZE
3560 015772 012701 035646    MOV      #WBUF,R1     ;SET START OF BUFFER
3561 015776 042721 140300    DS2:    BIC      #140300,(R1)+ ;MASK FOR 7 CH
3562 016002 005302          DEC      R2            ;SEE IF DONE
3563 016004 001374          BNE      DS2          ;IF NOT: BR
3564 016006 012702 000202    DS3:    MOV      #202,R2     ;R2=BUFFER SIZE +2
3565 016012 012701 036260    MOV      #RBUF,R1     ;R1=READ DATA START
3566 016016 005021    DS4:    CLR      (R1)+       ;CLEAR BUFFER
3567 016020 005302          DEC      R2            ;SEE IF DONE ALL
3568 016022 001375          BNE      DS4          ;IF NOT: BR
3569 016024 000207          RTS      PC            ;EXIT
3570
3571
3572
3573 016026 012701 177777    DAT1:   MOV      #-1,R1   ;R1=DATA
3574 016032 012702 000202    DAT1A:  MOV      #202,R2   ;R2=WORD COUNT +2
3575 016036 010123    DAT1B:  MOV      R1,(R3)+     ;LOAD BUFFER
3576 016040 005302          DEC      R2            ;SEE IF DONE
3577 016042 001375          BNE      DAT1B        ;IF NOT: BR
3578 016044 000167 177706    JMP      DS1          ;RETURN
3579
```

```
3580
3581
3582
3583 016050 005001
3584 016052 000167 177754
3585
3586
3587
3588 016056 012701 125125
3589 016062 000167 177744
3590
3591
3592
3593 016066 005001
3594 016070 012702 000404
3595 016074 110123
3596 016076 105201
3597 016100 005302
3598 016102 001374
3599 016104 000167 177646
3600

;ALL ZEROS*****
DAT2: CLR R1 ;R1=DATA
      JMP DAT1A ;LOAD BUFFER

;ONE/ZERO IN ALTERNATING CHARACTERS*****
DAT3: MOV #125125,R1 ;R1=DATA
      JMP DAT1A ;LOAD BUFFER

;ALL BITS 0-377*****
DAT4: CLR R1 ;R1=STARTING DATA
      MOV #404,R2 ;R2=CHARACTER COUNT
DAT4A: MOVB R1,(R3)+ ;LOAD BUFFER
      INCB R1 ;BUMP DATA
      DEC R2 ;SEE IF DONE
      BNE DAT4A ;IF NOT: BR
      JMP DS1 ;RETURN
```

```

3601
3602                                     ;WRAP AROUND MODE 0 GLOBAL*****
3603
3604 016110 012767 000006 162632 WAM0:  MOV    #6,WAM          ;SET WAM NUMBER
3605 016116 012767 000060 162626 WAM01: MOV    #60,FUN
3606 016124 005067 162624          CLR    DATC
3607 016130 012767 035646 162622          MOV    #WBUFF,DATAD ;SET BUFFER ADDRESS
3608 016136 012767 036260 162616          MOV    #RBUFF,RDAD  ;SET POINTER TO READ BUFFER
3609 016144 004767 000170          JSR    PC,SETUP     ;GO SET UP
3610 016150 000167 000512          JMP    EXEC
3611
3612                                     ;WRAP AROUND MODE 1 WRITE BUFFER*****
3613
3614 016154 012767 000010 162566 WAM1:  MOV    #10,WAM
3615 016162 000167 177730          JMP    WAM01
3616
3617                                     ;WRAP AROUND MODE 2 BIT FIDDLER WRITE*****
3618
3619 016166 012767 000012 162554 WAM2:  MOV    #12,WAM
3620 016174 012767 000060 162550          MOV    #60,FUN
3621 016202 005067 162546          CLR    DATC
3622 016206 012767 035646 162544          MOV    #WBUFF,DATAD
3623 016214 012767 036260 162540          MOV    #RBUFF,RDAD
3624 016222 004767 000112          WAM2A: JSR    PC,SETUP
3625 016226 000167 000434          JMP    EXEC
3626
3627                                     ;WRAP AROUND MODE 3 BIT FIDDLER READ*****
3628
3629 016232 012767 000014 162510 WAM3:  MOV    #14,WAM          ;SET WAM NUMBER
3630 016240 012767 000070 162504          MOV    #70,FUN          ;SET FUNCTION
3631 016246 012767 036260 162504          MOV    #RBUFF,DATAD   ;SET BUFFER ADDRESS
3632 016254 012767 035646 162474          MOV    #WBUFF,WTAD    ;SET POINTER TO WRITE BUFFER
3633 016262 005767 162520          TST    RDRVF
3634 016266 001411          BEQ    WAM3A
3635 016270 062767 000376 162462          ADD    #376,DATAD
3636 016276 062767 000377 162452          ADD    #377,WTAD
3637 016304 012767 000076 162440          MOV    #76,FUN          ;SET READ REVERSE CODE
3638 016312 032767 000020 162456 WAM3A: BIT    #20,UDES
3639 016320 001403          BEQ    WAM3B
3640 016322 016767 162462 162426          MOV    RCDP,WTAD
3641 016330 004767 000004          WAM3B: JSR    PC,SETUP     ;GO SET UP
3642 016334 000167 000326          JMP    EXEC          ;GO EXECUTE
3643

```



```

3644                                     ;REGISTER SETUP ROUTINE*****
3645
3646 016340 005767 162374      SETUP:  TST      STFLG      ;SEE IF SINGLE TEST
3647 016344 001403              BEQ      SET0      ;IF NOT: BR
3648 016346 005767 162426      TST      WPGFL     ;SEE IF HAVE SELECTED PATTERN
3649 016352 001002              BNE      SET1      ;IF SO: BR
3650 016354 004767 177260      SET0:  JSR      PC,DSUP ;GO DO DATA SETUP
3651 016360 004767 005622      SET1:  JSR      PC,INIT1 ;GO INIT SELECT DRIVE, SLAVE
3652 016364 004767 000664      JSR      PC,BOTT   ;GO ASSURE NOT AT BOT
3653 016370 012777 177400 162120  MOV      #-400,@FC ;SET FC=WCX2
3654 016376 032767 000020 162372  BIT      #20,UDES  ;SEE IF CORE DUMP
3655 016404 001403              BEQ      SET2      ;IF NOT: BR
3656 016406 012777 177000 162102  MOV      #-1000,@FC ;SET FC=WCX4
3657 016414 012777 177600 162070  SET2:  MOV      #-200,@WC ;SET WC
3658 016422 016777 162332 162064  MOV      DATAD,@BA ;SET BUS ADDRESS
3659 016430 032777 010000 162062  BIT      #10C00,@CS ;ASSURE DRIVE THERE
3660 016436 001417              BEQ      SP1       ;IF SO: BR
3661 016440 032777 020000 162122  BIT      #20000,@SWR ;SEE IF PRINT ERRORS
3662 016446 001004              BNE      SP01      ;IF NOT: BR
3663 016450 012704 035007      MOV      #WMSG4,R4
3664 016454 004767 006104      JSR      PC,TTOUT  ;PRINT NON-EXISTANT DRIVE
3665 016460 032777 100000 162102  SP01:  BIT      #100000,@SWR ;SEE IF HALT ON ERROR
3666 016466 001401              BEQ      SPO       ;IF NOT: BR
3667 016470 000000              HALT
3668 016472 000167 177662      SP0:   JMP      SET1      ;RESETUP
3669 016476 022767 000014 162244  SP1:   CMP      #14,WAM ;SEE IF WAM 3
3670 016504 001026              BNE      SP1B      ;IF NOT: BR
3671 016506 117767 162244 162240  MOVB     @WTAD,DATC ;GET FIRST CHAR
3672 016514 042767 177400 162232  BIC      #177400,DATC
3673 016522 000367 162226      SWAB     DATC
3674 016526 005767 162254      TST      RDRVF     ;SEE IF READ REVERSE
3675 016532 001403              BEQ      SP1A      ;IF NOT: BR
3676 016534 005367 162216      DEC      WTAD      ;DECREMENT POINTER
3677 016540 000410              BR       SP1B
3678 016542 005267 162210      SP1A:  INC      WTAD      ;BUMP POINTER
3679 016546 032767 000020 162222  BIT      #20,UDES  ;SEE IF CORE DUMP
3680 016554 001402              BEQ      SP1B      ;IF NOT: BR
3681 016556 005267 162174      INC      WTAD      ;BUMP POINTER AGAIN
3682 016562 056777 162210 161752  SP1B:  BIS      UDES,@TC  ;SET UNIT DESCRIPTION (DEN,PAR,FMT)
3683 016570 052777 000001 161736  BIS      #1,@MR    ;SET MAINT MODE
3684 016576 056777 162146 161730  BIS      WAM,@MR   ;SET WAM
3685 016604 056777 162144 161722  BIS      DATC,@MR  ;SET DATA
3686 016612 016777 162134 161670  MOV      FUN,@C1   ;SET FUNCTION
3687 016620 032777 040000 161674  BIT      #40000,@DS ;ASSURE NO ERROR
3688 016626 001001              BNE      SP3       ;IF NOT: BR
3689 016630 000207              RTS        ;RETURN
3690 016632 032777 020000 161730  SP3:  BIT      #20000,@SWR ;SEE IF PRINT ERRORS
3691 016640 001004              BNE      SP4       ;IF NOT: BR
3692 016642 012704 034750      MOV      #WMSG2,R4
3693 016646 004767 005712      JSR      PC,TTOUT  ;PRINT SETUP ERROR
3694 016652 032777 100000 161710  SP4:  BIT      #100000,@SWR ;SEE IF HALT ON ERROR
3695 016660 001401              BEQ      SP5       ;IF NOT: BR
3696 016662 000000              HALT
3697 016664 000207      SP5:  RTS        ;RETURN

```

```

3698                                     ;EXECUTE WAM ROUTINE*****
3699
3700 016666 000240          EXEC:  NOP
3701 016670 000240          NOP
3702 016672 032777 000040 161634  BIT    #40,@MR
3703 016700 001403          BEQ    EX0
3704 016702 042777 000040 161624  BIC    #40,@MR          ;ASSURE MAINT CLOCK IS ZERO
3705 016710 022767 000010 162032  EX0:  CMP    #10,WAM      ;IF NOT: CLEAR IT
3706 016716 003402          BLE    EX1          ;SEE IF WAM 1 OR 2 OR 3
3707 016720 000167 000372          JMP    EXW2        ;IF SO: BR
3708 016724 052777 000001 161556  EX1:  BIS    #1,@C1      ;GO DO WAM 0
3709 016732 005000          CLR    R0          ;SET GO BIT
3710 016734 012701 000002          MOV    #2,R1
3711 016740 032777 100000 161574  EX1A: BIT    #100000,@TC  ;SET DELAY
3712 016746 001404          BEQ    EX2          ;SEE IF ALPHA
3713 016750 005300          DEC    R0          ;IF SO: BR
3714 016752 001372          BNE    EX1A        ;AWAIT ALPHA
3715 016754 005301          DEC    R1
3716 016756 001370          BNE    EX1A
3717 016760 005077 161602          EX2:  CLR    @PSW
3718 016764 012701 000400          MOV    #400,R1      ;SET NUMBER OF CLKS
3719 016770 032767 000020 162000  BIT    #20,UDES     ;SEE IF CORE DUMP
3720 016776 001402          BEQ    EX3          ;IF NOT: BR
3721 017000 012701 001000          MOV    #1000,R1    ;SET CLOCKS LWCX4
3722 017004 022767 000014 161736  EX3:  CMP    #14,WAM   ;SEE IF WAM 3
3723 017012 001413          BEQ    EX5A        ;IF SO: BR
3724 017014 032767 002000 161754  BIT    #2000,UDES  ;SEE IF PE
3725 017022 001405          BEQ    EX5          ;IF NOT PE: BR
3726 017024 000241          CLC
3727 017026 006101          ROL    R1
3728 017030 062701 000246          ADD    #246,R1     ;SET TO ALLOW FOR PRE/POSTAMBLE
3729 017034 000402          BR    EX5A
3730 017036 062701 000010          EX5:  ADD    #10,R1  ;ADD CLOCKS FOR CRC AND LRC
3731 017042 022767 000014 161700  EX5A: CMP    #14,WAM   ;SEE IF WAM 3
3732 017050 001046          BNE    EX5C        ;IF NOT: BR
3733 017052 117700 161700          MOVB  @WTAD,R0
3734 017056 042700 177400          BIC    #177400,R0
3735 017062 005767 161720          TST   RDRVF
3736 017066 001403          BEQ    EX5A1       ;SEE IF REVERSE
3737 017070 005367 161662          DEC   WTAD        ;IF NOT: BR
3738 017074 000416          BR    EX5B        ;DEC POINTER
3739 017076 005267 161654          EX5A1: INC   WTAD
3740 017102 032767 000020 161666  BIT    #20,UDES     ;SEE IF CORE DUMP
3741 017110 001410          BEQ    EX5B        ;IF NOT: BR
3742 017112 005267 161640          INC   WTAD        ;BUMP POINTER
3743 017116 005777 161634          TST   @WTAD
3744 017122 001003          BNE    EX5B        ;SEE IF END
3745 017124 162767 000010 161624  SUB    #10,WTAD    ;IF NOT: BR
3746 017132 052777 000040 161374  EX5B: BIS    #40,@MR   ;RESTORE POINTER
3747 017140 017702 161370          MOV   @MR,R2      ;CLOCK UP
3748 017144 042702 177400          BIC   #177400,R2  ;READ MR
3749 017150 000300          SWAB R0           ;MASK OUT DATA
3750 017152 050002          BIS   R0,R2       ;POSITION DATA
3751 017154 010277 161354          MOV   R2,@MR      ;LOAD NEW DATA
3752 017160 000430          BR    EX5D        ;CLOCK DOWN AND LOAD NEW DATA
3753 017162 000240          NOP

```



```

3754 017164 000240      NOP
3755 017166 052777 000040 161340 EX5C:  BIS #40,@MR      ;CLOCK UP
3756 017174 042777 000040 161332      BIC #40,@MR      ;CLOCK DOWN
3757 017202 017700 161326      MOV @MR,R0       ;GET MR
3758 017206 000300      SWAB R0
3759 017210 032767 000010 161560      BIT #10,UDES     ;SEE IF EVEN PAR
3760 017216 001405      BEQ EX5C0        ;IF NOT: BR
3761 017220 010077 161536      MOV R0,@RDAD
3762 017224 005267 161532      INC RDAD
3763 017230 000402      BR EX5C1
3764 017232 110077 161524      EX5C0: MOVB R0,@RDAD ;PUT CHAR IN CORE
3765 017236 005267 161520      EX5C1: INC RDAD
3766 017242 000240      EX5D:  NOP
3767 017244 005301      DEC R1           ;SEE IF DONE CLKS
3768 017246 001275      BNE EX5A        ;IF NOT: BR
3769 017250 000167 003006      JMP EORP        ;GO DO EOR
3770
3771      ;ASSURE NOT AT BOT FOR WRAP TESTS*****
3772
3773 017254 032777 000002 161240 BOTT:  BIT #2,@DS     ;SEE IF BOT
3774 017262 001414      BEQ BOTTX       ;IF NOT: BR
3775 017264 052777 001700 161250      BIS #1700,@TC   ;SET NRZ
3776 017272 012777 000025 161210      MOV #25,@C1     ;DO ERASE
3777 017300 032777 000200 161214 BOTTA:  BIT #200,@DS
3778 017306 001774      BEQ BOTTA      ;AWAIT DRY
3779 017310 004767 004672      JSR PC,INIT1   ;INIT
3780 017314 000207      BOTTX: RTS PC   ;RETURN
3781

```



```

3782                                     :EXECUTE WAM 0*****
3783
3784 017316 000240                       EXW2:  NOP
3785 017320 016700 161210                 MOV    MR,R0           ;R0=MR
3786 017324 012703 000040                 MOV    #40,R3         ;R3=40
3787 017330 012767 017464 161332         MOV    #EXW2H,RTRN   ;SET INTERRUPT RETURN ADDRESS
3788 017336 012701 000200                 MOV    #200,R1       ;SET NUMBER OF CLOCKS = FC/2
3789 017342 032767 002000 161426         BIT    #2000,UDES    ;SEE IF PE
3790 017350 001402                       BEQ    EXW2A          ;IF NOT: BR
3791 017352 012701 000100                 MOV    #100,R1       ;ELSE SET CLKS = FC/4
3792 017356 016702 161400                 EXW2A: MOV   RDAD,R2   ;SET BUFFER ADDRESS
3793 017362 012777 000161 161120         MOV    #161,@C1      ;SET WRITE+GO
3794 017370 005077 161172                 CLR    @PSW          ;ALLOW INTERRUPT
3795 017374 030310                       EXW2B: BIT    R3,@R0
3796 017376 001776                       BEQ    EXW2B          ;AWAIT CLOCK UP
3797 017400 011022                       MOV    @R0,(R2)+     ;GET DATA
3798 017402 030310                       EXW2C: BIT    R3,@R0
3799 017404 001376                       BNE    EXW2C          ;AWAIT CLOCK DOWN
3800 017406 011022                       MOV    @R0,(R2)+     ;GET DATA
3801 017410 005301                       DEC    R1             ;SEE IF DONE ALL
3802 017412 001370                       BNE    EXW2B          ;IF NOT: BR
3803 017414 012701 000003                 EXW2E: MOV    #3,R1
3804 017420 005000                       CLR    R0             ;SET DELAY
3805 017422 005300                       EXW2F: DEC    R0
3806 017424 001376                       BNE    EXW2F
3807 017426 005301                       DEC    R1
3808 017430 001374                       BNE    EXW2F          ;DELAY
3809 017432 032777 020000 161130         BIT    #20000,@SWR   ;SEE IF ERROR PRINT
3810 017440 001004                       BNE    EXW2G          ;IF NOT: BR
3811 017442 012704 035212                 MOV    #WMSG24,R4
3812 017446 004767 005112                 JSR    PC,TTOUT      ;PRINT NO INTERUPT
3813 017452 032777 100000 161110         EXW2G: BIT    #100000,@SWR ;SEE IF HALT ON ERROR
3814 017460 001401                       BEQ    EXW2H          ;IF NOT: BR
3815 017462 000000                       HALT
3816 017464 012701 036260                 EXW2H: MOV    #RBUFF,R1 ;GET START OF READ BUFFER
3817 017470 012700 000400                 MOV    #400,R0       ;SET SIZE
3818 017474 010102                       MOV    R1,R2
3819 017476 012203                       EXW2J: MOV    (R2)+,R3
3820 017500 000303                       SWAB   R3
3821 017502 032767 000010 161266         BIT    #10,UDES      ;SEE IF EVEN PAR
3822 017510 001402                       BEQ    EXW2J0         ;IF NOT: BR
3823 017512 010321                       MOV    R3,(R1)+     ;SAVE PAR + DATA
3824 017514 000401                       BR     EXW2J1
3825 017516 110321                       EXW2J0: MOVB  R3,(R1)+ ;ASSEMBLE DATA IN BYTES
3826 017520 005300                       EXW2J1: DEC    R0
3827 017522 001365                       BNE    EXW2J         ;CONTINUE FOR ALL
3828 017524 032777 000200 161036         BIT    #200,@SWR    ;SEE IF STATUS CHECK
3829 017532 001002                       BNE    EXW2K         ;IF NOT: BR
3830 017534 004767 000022                 JSR    PC,WSTCK      ;ELSE GO CHECK STATUS
3831 017540 000240                       EXW2K: NOP
3832 017542 032777 000400 161020         BIT    #400,@SWR    ;SEE IF DATA CHECK
3833 017550 001002                       BNE    EXW2X         ;IF NOT: BR
3834 017552 004767 000276                 JSR    PC,DCHK       ;ELSE GO CHECK DATA
3835 017556 000240                       EXW2X: NOP
3836 017560 000207                       RTS    PC             ;EXIT

```

```

3837
3838
3839
3840 017562 000240
3841 017564 005067 161202
3842 017570 005067 161024
3843 017574 012767 035034 161070
3844 017602 017702 160702
3845 017606 016705 161206
3846 017612 004767 000112
3847 017616 012767 035061 161046
3848 017624 017702 160670
3849 017630 016705 161166
3850 017634 056705 160764
3851 017640 004767 000064
3852 017644 012767 035067 161020
3853 017652 017702 160644
3854 017656 016705 161142
3855 017662 004767 000042
3856 017666 012767 035074 160776
3857 017674 017702 160624
3858 017700 016705 161122
3859 017704 004767 000020
3860 017710 005767 161056
3861 017714 001456
3862 017716 005777 160646
3863 017722 100053
3864 017724 000000
3865 017726 000451
3866 017730 000240
3867 017732 020205
3868 017734 001446
3869 017736 005267 161030
3870 017742 032777 020000 160620
3871 017750 001040
3872 017752 005767 160642
3873 017756 001010
3874 017760 016704 160636
3875 017764 004767 004574
3876 017770 012704 035176
3877 017774 004767 004564
3878 020000 012767 000001 160612
3879 020006 016704 160660
3880 020012 004767 004546
3881 020016 012704 026200
3882 020022 004767 004536
3883 020026 010503
3884 020030 004767 004656
3885 020034 012704 026207
3886 020040 004767 004520
3887 020044 010203
3888 020046 004767 004640
3889 020052 000207
3890

;WRAP AROUND STATUS CHECK ROUTINE*****
WSTCK: NOP
CLR SERFL ;CLEAR ERROR FLAG
CLR HDRFL ;CLEAR HEADER FLAG
MOV #WMSG6,ERADD ;SET CODE=CS1
MOV @C1,R2 ;GET RCVD CS1
MOV WCS1,R5 ;GET EXPT CS1
JSR PC,WSTG ;GO CHK
MOV #WMSG6D,ERADD ;SET CODE=CS2
MOV @CS,R2 ;SET RCVD CS2
MOV WCS2,R5 ;GET EXPT CS2
BIS DRVN,R5 ;SET DRIVE NUMBER IN EXPT CS2
JSR PC,WSTG ;GO CHK
MOV #WMSG6E,ERADD ;SET CODE=DS
MOV @DS,R2 ;SET RCVD DS
MOV WDS,R5 ;GET EXPT DS
JSR PC,WSTG ;GO CHK
MOV #WMSG6F,ERADD ;SET CODE=ER
MOV @ER,R2 ;GET RCVD ER
MOV WER,R5 ;GET EXPT ER
JSR PC,WSTG ;GO CHK
TST SERFL ;SEE IF ANY ERRORS
BEQ WSTX ;IF NOT: BR
TST @SWR ;SEE IF HALT ON ERROR
BPL WSTX ;IF NOT: BR
HALT
BR WSTX ;CONTINUE
WSTG: NOP
CMP R2,R5 ;SEE IF EXPT=RCVD
BEQ WSTX ;IF SO: BR
INC SERFL ;SET ERROR FLAG
BIT #20000,@SWR ;SEE IF PRINT ERRORS
BNE WSTX ;IF NOT: BR
TST HDRFL ;SEE IF DONE HEADER
BNE WSTG0 ;IF SO: BR
MOV EMADDR,R4 ;PRINT TEST HEADER
JSR PC,TTOUT
MOV #WMSG23,R4
JSR PC,TTOUT ;PRINT STATUS TAG
MOV #1,HDRFL ;SET HEADER FLAG
MOV ERADD,R4
JSR PC,TTOUT ;PRINT CODE
MOV #MSG12,R4
JSR PC,TTOUT ;PRINT EXPT TAG
MOV R5,R3
JSR PC,OCTP ;PRINT EXPT STATUS
MOV #MSG13,R4
JSR PC,TTOUT ;PRINT RCVD TAG
MOV R2,R3
JSR PC,OCTP ;PRINT RCVD STATUS
WSTX: RTS PC ;RETURN

```



```

3891
3892
3893                                     ;DATA CHECK ROUTINE*****
3894 020054 000240          DCHK:  NOP
3895 020056 005067 160536    CLR      HDRFL          ;CLEAR HEADER FLAG
3896 020062 005067 160700    CLR      DERFL          ;CLEAR DATA ERROR FLAG
3897 020066 005067 160702    CLR      CRCNT          ;CLEAR CHAR CNTR
3898 020072 032767 000010 160676  BIT      #10,UDES        ;SEE IF EVEN PARITY
3899 020100 001402          BEQ      DCHKA0          ;IF NOT: BR
3900 020102 000167 000600    JMP      DCHKE          ;ELSE GO CHECK EVEN
3901 020106 022767 000010 160634 DCHKA0: CMP      #10,WAM        ;SEE IF WAM 1
3902 020114 001006          BNE      DCHKA          ;IF NOT: BR
3903 020116 032767 002000 160652  BIT      #2000,UDES      ;SEE IF PE
3904 020124 001402          BEQ      DCHKA          ;IF NOT: BR
3905 020126 000167 001220    JMP      PRCHK          ;GO CHK DATA
3906 020132 012700 177400    DCHKA:  MOV      #-400,RO  ;SET NUMBER OF CHARACTERS
3907 020136 022767 000012 160604  CMP      #12,WAM
3908 020144 001006          BNE      DCHKA1          ;IF NOT WRAP 2: BR
3909 020146 032767 000020 160622  BIT      #20,UDES
3910 020154 001402          BEQ      DCHKA1          ;IF NOT CORE DUMP: BR
3911 020156 012700 177000    MOV      #-1000,RO
3912 020162 022767 000006 160560 DCHKA1: CMP      #6,WAM
3913 020170 001007          BNE      DCHKA2
3914 020172 032767 002000 160576  BIT      #2000,UDES      ;SEE IF PE MODE
3915 020200 001430          BEQ      DCHKB          ;IF NOT: BR
3916 020202 012700 177600    MOV      #-200,RO       ;SET CHAR CNTR TO FC/2 FOR PE
3917 020206 000425          BR       DCHKB
3918 020210 022767 000012 160532 DCHKA2: CMP      #12,WAM      ;SEE IF WRAP 2
3919 020216 001021          BNE      DCHKB          ;IF NOT: BR
3920 020220 032767 002000 160550  BIT      #2000,UDES      ;SEE IF PE
3921 020226 001415          BEQ      DCHKB          ;IF NOT: BR
3922 020230 012700 177653    MOV      #-125,RO       ;POINT TO START OF DATA
3923 020234 012767 000001 160522  MOV      #1,W2FLG        ;SET WRAP 2 FLAG
3924 020242 004767 000014    JSR      PC,DCHKB        ;GO CHECK DATA
3925 020246 004767 001400    JSR      PC,W1DCHK       ;GO CHECK WRAP 1 DATA
3926 020252 005067 160506    CLR      W2FLG
3927 020256 000167 000362    JMP      DCHKX
3928 020262 005067 160500          DCHKB:  CLR      DERFL
3929 020266 012701 035646    MOV      #WBUFF,R1      ;SET GOOD POINTER
3930 020272 012702 036260    MOV      #RBUFF,R2      ;SET READ POINTER
3931 020276 032767 000020 160472  BIT      #20,UDES        ;SEE IF CORE DUMP
3932 020304 001416          BEQ      DCHKO          ;IF NOT: BR
3933 020306 022767 000012 160434  CMP      #12,WAM        ;SEE IF WAM 2
3934 020314 001011          BNE      DCHKD          ;IF NOT: BR
3935 020316 005767 160460    TST      PATRN          ;SEE IF PATTERN 0
3936 020322 001003          BNE      DCHKC          ;IF NOT: BR
3937 020324 012701 001054    MOV      #WCDPO,R1      ;SET CORE DUMP PATTERN 0
3938 020330 000404          BR       DCHKO          ;GO CHECK DATA
3939 020332 012701 001042    DCHKC:  MOV      #WCDP2,R1 ;SET CORE DUMP WRITE PATTERN 2
3940 020336 000401          BR       DCHKO          ;GO CHECK DATA
3941 020340 000240          DCHKD:  NOP
3942 020342 121112          DCHKO:  CMPB      (R1),(R2) ;SEE IF DATA OK
3943 020344 001466          BEQ      DCHK2          ;IF SO: BR
3944 020346 032777 020000 160214  BIT      #20000,@SWR     ;SEE IF PRINT ERRORS
3945 020354 001062          BNE      DCHK2          ;IF NOT: BR
3946 020356 005767 160236    TST      HDRFL          ;SEE IF DONE HEADER

```



```

3947 020362 001004          BNE      DCHK1          ;IF SO: BR
3948 020364 016704 160232    MOV      EMADDR,R4
3949 020370 004767 004170    JSR      PC,TTOUT      ;PRINT HEADER
3950 020374 005767 160366    DCHK1:  TST      DERFL   ;SEE IF FIRST ERROR
3951 020400 001014          BNE      DCHK1A        ;IF NOT: BR
3952 020402 012704 035144    MOV      #WMSG16,R4
3953 020406 004767 004152    JSR      PC,TTOUT      ;PRINT DATA ERROR TAG
3954 020412 012704 035371    MOV      #WMSG32,R4
3955 020416 004767 004142    JSR      PC,TTOUT      ;PRINT PATRN TAG
3956 020422 016703 160354    MOV      PATRN,R3
3957 020426 004767 004260    JSR      PC,OCTP       ;PRINT PATTERN NUMBER
3958 020432 012767 000001 160160 DCHK1A: MOV      #1,HDRFL  ;SET HEADER FLAG
3959 020440 012767 000001 160320    MOV      #1,DERFL     ;SET DATA ERROR FLAG
3960 020446 012704 035170    MOV      #WMSG21,R4
3961 020452 004767 004106    JSR      PC,TTOUT      ;PRINT CHARACTER NUMBER TAG
3962 020456 016703 160312    MOV      CRCNT,R3
3963 020462 004767 004224    JSR      PC,OCTP       ;PRINT CHARACTER NUMBER
3964 020466 012704 035156    MOV      #WMSG17,R4
3965 020472 004767 004066    JSR      PC,TTOUT      ;PRINT GOOD TAG
3966 020476 111103          MOV      (R1),R3
3967 020500 004767 004434    JSR      PC,DOUT       ;PRINT GOOD DATA
3968 020504 012704 035163    MOV      #WMSG20,R4
3969 020510 004767 004050    JSR      PC,TTOUT      ;PRINT BAD TAG
3970 020514 111203          MOV      (R2),R3
3971 020516 004767 004416    JSR      PC,DOUT       ;PRINT BAD DATA
3972 020522 005767 160236    DCHK2:  TST      W2FLG
3973 020526 001020          BNE      DCHK2B
3974 020530 005201          INC      R1             ;BUMP POINTER
3975 020532 032767 000020 160236    BIT      #20,UDES      ;SEE IF CORE DUMP
3976 020540 001413          BEQ      DCHK2B        ;IF NOT: BR
3977 020542 022767 000012 160200    CMP      #12,WAM       ;SEE IF WAM 2
3978 020550 001006          BNE      DCHK2A        ;IF NOT: BR
3979 020552 005201          INC      R1             ;BUMP POINTER
3980 020554 005711          TST      (R1)          ;SEE IF END OF PATTERN
3981 020556 001004          BNE      DCHK2B        ;IF NOT: BR
3982 020560 162701 000010    SUB      #10,R1        ;RESET POINTER TO START OF PATTERN
3983 020564 000401          BR       DCHK2B        ;CONTINUE CHECK
3984 020566 000240          DCHK2A: NOP
3985 020570 005202          DCHK2B: INC      R2
3986 020572 022767 000006 160150    CMP      #6,WAM        ;SEE IF WAM 0
3987 020600 001005          BNE      DCHK3
3988 020602 032767 002000 160166    BIT      #2000,UDES    ;SEE IF PE
3989 020610 001401          BEQ      DCHK3        ;IF NOT PE: BR
3990 020612 005201          INC      R1             ;BUMP WRITE DATA ADDRESS
3991 020614 005267 160154          DCHK3: INC      CRCNT   ;BUMP CHAR CNTR
3992 020620 032777 000400 157742    BIT      #400,@SWR     ;SEE IF CONT DATA CHK
3993 020626 001006          BNE      DCHKX        ;IF NOT: BR
3994 020630 005200          INC      R0             ;SEE IF DONE
3995 020632 001243          BNE      DCHK0        ;IF NOT: BR
3996 020634 005767 160124    TST      W2FLG
3997 020640 001401          BEQ      DCHKX
3998 020642 000207          RTS      PC
3999 020644 032777 100000 157716 DCHKX: BIT      #100000,@SWR ;SEE IF HALT ON ERROR
4000 020652 001404          BEQ      DCHKX1       ;IF NOT: BR
4001 020654 005767 160106    TST      DERFL        ;SEE IF DATA ERROR OCCURED
4002 020660 001401          BEQ      DCHKX1       ;IF NOT: BR

```

4003	020662	000000					
4004	020664	005067	160104	DCHKX1:	CLR	CRCNT	:CLEAR CHAR CNTR
4005	020670	005067	157724		CLR	HDRFL	:CLEAR HEADER FLAG
4006	020674	005067	160066		CLR	DERFL	:CLEAR DATA ERROR FLAG
4007	020700	005067	160064		CLR	PREFL	:CLEAR PREAMBLE FLAG
4008	020704	000207			RTS	PC	:RETURN



```

4009
4010                                     ;EVEN PARITY DATA CHECK*****
4011
4012 020706 000240          DCHKE:  NOP
4013 020710 012700 177400      MOV     #-400,R0          ;SET NUMBER OF CHARACTERS
4014 020714 012701 035646      MOV     #WBUF,R1         ;R1=START OF WRITE BUFFER
4015 020720 012702 036260      MOV     #RBUF,R2         ;R2=START OF READ BUFFER
4016 020724 111105          DCKE0: MOVB    (R1),R5      ;GET EXPT DATA
4017 020726 005003          CLR     R3
4018 020730 012704 000010      MOV     #10,R4           ;SET NUMBER OF BITS
4019 020734 032705 000001      DCKE1: BIT     #1,R5       ;SEE IF ONE BIT
4020 020740 001401          BEQ     DCKE2            ;IF NOT: BR
4021 020742 005203          INC     R3               ;COUNT ONE BITS FOR PARITY CHECK
4022 020744 005304          DCKE2: DEC     R4           ;SEE IF DONE
4023 020746 001402          BEQ     DCKE3            ;IF SO: BR
4024 020750 006005          ROR     R5               ;POINT TO NEXT BIT
4025 020752 000770          BR      DCKE1
4026 020754 000240          DCKE3: NOP
4027 020756 111105          MOVB    (R1),R5          ;GET EXPT DATA
4028 020760 042705 177400      BIC     #177400,R5       ;MASK DATA FIELD
4029 020764 005703          TST     R3
4030 020766 001003          BNE     DCKE4            ;IF NO ONE BITS SET: BR
4031 020770 012705 100020      MOV     #100020,R5
4032 020774 000405          BR      DCKE5
4033 020776 032703 000001      DCKE4: BIT     #1,R3       ;SEE IF ODD NUMBER OF ONE BITS
4034 021002 001402          BEQ     DCKE5            ;IF NOT: BR
4035 021004 052705 100000      BIS     #100000,R5       ;SET EVEN PARITY BIT=1
4036 021010 042712 077400      DCKE5: BIC     #77400,(R2) ;MASK DATA FIELD
4037 021014 020512          CMP     R5,(R2)         ;SEE IF DATA + PARITY GOOD
4038 021016 001477          BEQ     DCKE10          ;IF SO: BR
4039 021020 032777 020000 157542 BIT     #20000,@SWR      ;SEE IF ERROR PRINT
4040 021026 001073          BNE     DCKE10          ;IF NOT: BR
4041 021030 005767 157564      TST     HDRFL           ;SEE IF DONE HEADER
4042 021034 001004          BNE     DCKE6            ;IF SO: BR
4043 021036 016704 157560      MOV     EMADDR,R4
4044 021042 004767 003516      JSR     PC,TTOUT         ;PRINT HEADER
4045 021046 005767 157714      DCKE6: TST     DERFL       ;SEE IF FIRST BAD CHAR
4046 021052 001014          BNE     DCKE7            ;IF NOT: BR
4047 021054 012704 035144      MOV     #WMSG16,R4
4048 021060 004767 003500      JSR     PC,TTOUT         ;PRINT BAD DATA TAG
4049 021064 012704 035371      MOV     #WMSG32,R4
4050 021070 004767 003470      JSR     PC,TTOUT         ;PRINT PATTERN TAG
4051 021074 016703 157702      MOV     PATRN,R3
4052 021100 004767 003606      JSR     PC,OCTP         ;PRINT PATTERN NUMBER
4053 021104 000240          DCKE7: NOP
4054 021106 012767 000001 157652 MOV     #1,DERFL         ;SET DATA ERROR FLAG
4055 021114 012767 000001 157476 MOV     #1,HDRFL         ;SET HEADER FLAG
4056 021122 012704 035170      MOV     #WMSG21,R4
4057 021126 004767 003432      JSR     PC,TTOUT         ;PRINT CHAR NUMBER TAG
4058 021132 016703 157636      MOV     CRCNT,R3
4059 021136 004767 003550      JSR     PC,OCTP         ;ORINT CHAR NUMBER
4060 021142 012704 035156      MOV     #WMSG17,R4
4061 021146 004767 003412      JSR     PC,TTOUT         ;PRINT GOOD DATA TAG
4062 021152 110503          MOVB    R5,R3
4063 021154 004767 003760      JSR     PC,DOUT         ;PRINT EXPT DATA
4064 021160 010503          MOV     R5,R3

```



```

4065 021162 004767 000070      JSR    PC,DCKEP      ;GO PRINT PARITY BIT
4066 021166 000240      NOP
4067 021170 012704 035163      MOV    #WMSG20,R4
4068 021174 004767 003364      JSR    PC,TTOUT      ;PRINT BAD TAG
4069 021200 111203      MOVB   (R2),R3
4070 021202 004767 003732      JSR    PC,DOUT       ;PRINT BAD DATA
4071 021206 011203      MOV    (R2),R3
4072 021210 004767 000042      JSR    PC,DCKEP      ;GO PRINT PARITY BIT
4073 021214 000240      NOP
4074 021216 005201      DCKE10: INC    R1
4075 021220 005722      TST    (R2)+         ;BUMP POINTERS
4076 021222 005267 157546      INC    CRCNT         ;BUMP CHAR CNTR
4077 021226 032777 000400 157334      BIT    #400,@SWR     ;SEE IF CONTINUE DATA CHECK
4078 021234 001402      BEQ    DCKE11        ;IF SO: BR
4079 021236 000167 177402      JMP    DCHKX         ;GO TO END OF DATA CHECK
4080 021242 005200      DCKE11: INC    R0     ;SEE IF DONE
4081 021244 001402      BEQ    DCKE12        ;IF SO: BR
4082 021246 000167 177452      JMP    DCKE0         ;ELSE CONTINUE
4083 021252 000167 177366      DCKE12: JMP    DCHKX  ;GO TO END OF DATA CHECK
4084 021256 000240      DCKEP:  NOP
4085 021260 012767 000240 157326      MOV    #240,TOB
4086 021266 004767 003372      JSR    PC,TOG        ;SPACE
4087 021272 012767 000260 157314      MOV    #260,TOB     ;SET PAR=0
4088 021300 005703      TST    R3            ;SEE IF PARITY REALLY=0
4089 021302 100002      BPL    DCKEPO        ;IF SO: BR
4090 021304 005267 157304      INC    TOB           ;ELSE SET TO 1
4091 021310 004767 003350      DCKEPO: JSR   PC,TOG ;PRINT PARITY BIT
4092 021314 000207      RTS    PC            ;RETURN
4093
    
```

```

4094
4095
4096
4097 021316 012700 000051      PSCHK: MOV #51,R0      ;SET SIZE OF POSTAMBLE
4098 021322 012701 035524      MOV #POST,R1      ;SET POINTER TO POSTAMBLE
4099 021326 005067 157266      CLR HDRFL        ;CLEAR HEADER FLAG
4100 021332 005067 157436      CLR CRCNT       ;CLEAR CHAR CNTR
4101 021336 005067 157424      CLR DERFL       ;CLEAR DATA ERROR FLAG
4102 021342 000240
4103 021344 000240
4104 021346 000167 000016      JMP PDO         ;GO CHECK POSTAMBLE
4105
4106 021352 012700 000051      PRCHK: MOV #51,R0      ;SET SIZE OF PREAMBLE
4107 021356 012701 035402      MOV #PRE,R1      ;SET POINTER TO PREAMBLE
4108 021362 012702 036260      MOV #RBUF,R2     ;SET POINTER TO START OF READ BUFFER
4109 021366 022122              CMP (R1)+,(R2)+  ;BUMP ADDRESS POINTERS
4110 021370 121112              PDO:  CMPB (R1),(R2) ;CHECK DATA
4111 021372 001004              BNE PD1         ;IF NOT GOOD: BR
4112 021374 126162 000001 000001  CMPB 1(R1),1(R2) ;COMPARE COMPLIMENT BYTE
4113 021402 001477              BEQ PD5         ;IF GOOD: BR
4114 021404 032777 020000 157156 PD1:  BIT #20000,@SWR  ;SEE IF PRINT INHIBIT
4115 021412 001073              BNE PD5         ;IF SO: BR
4116 021414 005767 157200      TST HDRFL       ;SEE IF DONE HEADER
4117 021420 001020              BNE PD4         ;IF SO: BR
4118 021422 016704 157174      MOV EMADDR,R4
4119 021426 004767 003132      JSR PC,TTOUT    ;PRINT TEST HEADER
4120 021432 005767 157332      TST PREFL      ;SEE IF PREAMBLE CHECK
4121 021436 001403              BEQ PD2         ;IF NOT: BR
4122 021440 012704 035315      MOV #WMSG29,R4  ;SET POSTAMBLE HEADER
4123 021444 000402              BR PD3
4124 021446 012704 035277      PD2:  MOV #WMSG28,R4 ;SET PREAMBLE HEADER
4125 021452 004767 003106      PD3:  JSR PC,TTOUT ;PRINT HEADER
4126 021456 005267 157136      INC HDRFL
4127 021462 012704 035170      PD4:  MOV #WMSG21,R4
4128 021466 004767 003072      JSR PC,TTOUT    ;PRINT CHAR NUMBER TAG
4129 021472 016703 157276      MOV CRCNT,R3
4130 021476 004767 003210      JSR PC,OCTP     ;PRINT CHAR NUMBER
4131 021502 012704 035156      MOV #WMSG17,R4
4132 021506 004767 003052      JSR PC,TTOUT    ;PRINT GOOD TAG
4133 021512 116103 000001      MOVB 1(R1),R3
4134 021516 004767 003416      JSR PC,DOUT     ;PRINT GOOD CHAR
4135 021522 012767 000240 157064  MOV #240,TOB
4136 021530 004767 003130      JSR PC,TOG
4137 021534 111103      MOVB (R1),R3
4138 021536 004767 003376      JSR PC,DOUT     ;PRINT COMPLIMENT
4139 021542 012704 035163      MOV #WMSG20,R4
4140 021546 004767 003012      JSR PC,TTOUT    ;PRINT BAD TAG
4141 021552 116203 000001      MOVB 1(R2),R3
4142 021556 004767 003356      JSR PC,DOUT     ;PRINT BAD CHAR
4143 021562 012767 000240 157024  MOV #240,TOB
4144 021570 004767 003070      JSR PC,TOG
4145 021574 111203      MOVB (R2),R3
4146 021576 004767 003336      JSR PC,DOUT     ;PRINT COMPLIMENT
4147 021602 022122      PD5:  CMP (R1)+,(R2)+  ;BUMP ADDRESS POINTERS
4148 021604 005267 157164      INC CRCNT       ;BUMP CHAR NUMBER
4149 021610 005300      DEC RO          ;SEE IF DONE

```

4150 021612 001266  
4151 021614 005767 157150  
4152 021620 001402  
4153 021622 000167 177016  
4154 021626 005267 157136  
4155 021632 005067 156762  
4156 021636 005067 157132  
4157 021642 005067 157120  
4158 021646 000167 000000  
4159

PD6:

BNE	PDO	;IF NOT: BR
TST	PREFL	;SEE IF PREAMBLE
BEQ	PD6	;IF SO: BR
JMP	DCHKX	;GO TO EXIT ROUTINE
INC	PREFL	;SET PREAMBLE FLAG
CLR	HDRFL	;CLEAR HEADER FLAG
CLR	CRCNT	;CLEAR CHAR CNTR
CLR	DERFL	;CLEAR DATA ERROR FLAG
JMP	WIDCHK	;GO CHECK WRAP 1 DATA



```

4160
4161
4162
4163 021652 012700 177400
4164 021656 012701 035646
4165 021662 012702 036260
4166 021666 062702 000124
4167 021672 005767 157066
4168 021676 001401
4169 021700 005302
4170 021702 111105
4171 021704 120512
4172 021706 001007
4173 021710 005767 157050
4174 021714 001001
4175 021716 105105
4176 021720 120562 000001
4177 021724 001510
4178 021726 032777 020000 156634
4179 021734 001104
4180 021736 005767 156656
4181 021742 001020
4182 021744 016704 156652
4183 021750 004767 002610
4184 021754 012704 035144
4185 021760 004767 002600
4186 021764 012704 035371
4187 021770 004767 002570
4188 021774 016703 157002
4189 022000 004767 002706
4190 022004 012767 000001 156606
4191 022012 012704 035170
4192 022016 004767 002542
4193 022022 016703 156746
4194 022026 004767 002660
4195 022032 012704 035156
4196 022036 004767 002522
4197 022042 111105
4198 022044 110503
4199 022046 005767 156712
4200 022052 001001
4201 022054 105103
4202 022056 004767 003056
4203 022062 012767 000240 156524
4204 022070 004767 002570
4205 022074 110503
4206 022076 004767 003036
4207 022102 012704 035163
4208 022106 004767 002452
4209 022112 116203 000001
4210 022116 004767 003016
4211 022122 012767 000240 156464
4212 022130 004767 002530
4213 022134 111203
4214 022136 004767 002776
4215 022142 005267 156620

```

;WAM 1 PE DATA CHECK\*\*\*\*\*

```

W1DCHK: MOV #-400,R0 ;SET NUMBER OF CHAR TO CHECK
MOV #WUBUFF,R1 ;SET WRITE DATA POINTER
MOV #RUBUFF,R2 ;SET READ DATA POINTER
ADD #124,R2 ;POINT TO START OF DATA
TST W2FLG ;SEE IF WRAP 2
BEQ W1D0 ;IF NOT WAM 2: BR
DEC R2 ;RESET POINTER

W1D0: MOV (R1),R5
CMPB R5,(R2) ;CHECK DATA
BNE W1D1 ;IF NOT GOOD:BR
TST W2FLG ;SEE IF WRAP 2
BNE W1D0A ;IF SO: BR
COMB R5 ;COMPLIMENT EXPT DATA

W1D0A: CMPB R5,1(R2) ;CHECK COMPLIMENT DATA
BEQ W1D3 ;IF GOOD: BR

W1D1: BIT #20000,ASWR ;SEE IF PRINT INHIBIT
BNE W1D3 ;IF SO: BR
TST HDRFL ;SEE IF DONE HEADER
BNE W1D2 ;IF SO: BR

MOV EMADDR,R4
JSR PC,TTOUT ;PRINT TEST HEADER
MOV #WMSG16,R4
JSR PC,TTOUT ;PRINT BAD DATA TAG
MOV #WMSG32,R4
JSR PC,TTOUT ;PRINT PATRN TAG
MOV PATRN,R3
JSR PC,OCTP ;PRINT PATTERN NUMBER

W1D2: MOV #1,HDRFL ;SET HEADER FLAG
MOV #WMSG21,R4
JSR PC,TTOUT ;PRINT CHAR NUMBER TAG
MOV CRCNT,R3
JSR PC,OCTP ;PRINT CHAR NUMBER
MOV #WMSG17,R4
JSR PC,TTOUT ;PRNT GOOD TAG
MOVB (R1),R5
MOVB R5,R3 ;GET GOOD CHAR
TST W2FLG ;SEE IF WRAP 2
BNE W1D2A ;IF SO: BR
COMB R3 ;ELSE COMPLIMENT CHAR

W1D2A: JSR PC,DOUT ;PRINT CHARACTER
MOV #240,TOB
JSR PC,TOG ;SPACE
MOVB R5,R3
JSR PC,DOUT ;PRINT CHAR
MOV #WMSG20,R4
JSR PC,TTOUT ;PRINT BAD TAG
MOVB 1(R2),R3
JSR PC,DOUT ;PRINT BAD CHAR
MOV #240,TOB
JSR PC,TOG ;SPACE
MOVB (R2),R3
JSR PC,DOUT ;PRINT CHAR
INC DERFL ;SET DATA ERROR FLAG

```

```

4216 022146 122122
4217 022150 105722
4218 022152 005267 156616
4219 022156 000406
4220 022160 005767 156600
4221 022164 001401
4222 022166 000207
4223 022170 000167 177122
4224 022174 005200
4225 022176 001770
4226 022200 000167 177476
4227
4228
4229
4230 022204 000240
4231 022206 012700 000050
4232 022212 012701 035402
4233 022216 005721
4234 022220 012721 177400
4235 022224 005300
4236 022226 001374
4237 022230 012701 035524
4238 022234 012700 000050
4239 022240 012721 000377
4240 022244 012721 177400
4241 022250 005300
4242 022252 001374
4243 022254 000207
    
```

```

W1D3:  CMPB  (R1)+,(R2)+  ;BUMP ADDRESS
        TSTB  (R2)+      ;BUMP ADDRESS
        INC   CRCNT      ;BUMP CHAR CNTR
        BR    W1D5
W1D4:  TST   W2FLG       ;SEE IF WRAP 2
        BEQ   W1D4A      ;IF NOT: BR
        RTS   PC         ;ELSE RETURN
W1D4A: JMP   PSCHK       ;GO CHECK POSTAMBLE
W1D5:  INC   RO
        BEQ   W1D4
        JMP   W1D0
    
```

;PREAMBLE/POSTAMBLE GENERATE SUBROUTINE\*\*\*\*\*

```

PPGEN: NOP
        MOV   #50,R0      ;SET SIZE OF PREAMBLE
        MOV   #PRE,R1
        TST   (R1)+      ;SET ADDRESS OF PRE
PPGO:  MOV   #177400,(R1)+ ;FILL TABLE
        DEC   RO         ;SEE IF DONE
        BNE   PPG0       ;IF NOT: BR
        MOV   #POST,R1   ;SET ADDRESS OF POST
        MOV   #50,R0     ;SET SIZE OF POST
        MOV   #377,(R1)+ ;SET SYNC CHAR
PPG1:  MOV   #177400,(R1)+ ;FILL TABLE
        DEC   RO         ;SEE IF DONE
        BNE   PPG1       ;IF NOT: BR
        RTS   PC         ;RETURN
    
```



```

4244
4245
4246
4247 022256 005267 156414      EORPA:  INC      TEMP2      ;SET WRAP FLAG
4248 022262 017700 156246      EORP:   MOV      @MR,R0     ;GET MAINT REG
4249 022266 042700 000036      BIC     #36,R0     ;CLEAR CURRENT OP CODE
4250 022272 052700 000024      BIS     #24,R0     ;SET EOR CLEAR OP CODE
4251 022276 010077 156232      MOV     R0,@MR     ;DO EOR
4252 022302 042777 000037 156224      BIC     #37,@MR     ;CLEAR EOR AND MM
4253 022310 005000
4254 022312 012701 000002      MOV     #2,R1
4255 022316 032777 000001 156164      EORP1:  BIT      #1,@C1     ;SEE IF GO GONE
4256 022324 001430      BEQ     EORP2      ;IF SO: BR
4257 022326 005300      DEC     R0
4258 022330 001372      BNE     EORP1      ;AWAIT GO RESET
4259 022332 005301      DEC     R1
4260 022334 001370      BNE     EORP1
4261 022336 032777 020000 156224      BIT     #20000,@SWR ;SEE IF ERROR PRINT INHIBIT
4262 022344 001020      BNE     EORP2      ;IF SO: BR
4263 022346 005767 156246      TST     HDRFL      ;SEE IF DONE HEADER
4264 022352 001004      BNE     EORP1A     ;IF SO: BR
4265 022354 016704 156242      MOV     EMADDR,R4
4266 022360 004767 002200      JSR     PC,TTOUT   ;PRINT HEADER
4267 022364 012704 035334      EORP1A: MOV     #WMSG31,R4
4268 022370 004767 002170      JSR     PC,TTOUT   ;PRINT EOR GO BIT ERROR
4269 022374 032777 100000 156166      BIT     #100000,@SWR ;SEE IF HALT ON ERROR
4270 022402 001401      BEQ     EORP2      ;IF NOT: BR
4271 022404 000000      HALT
4272 022406 000240      EORP2:  NOP
4273 022410 005767 156262      TST     TEMP2      ;SEE IF WAM
4274 022414 001015      BNE     EORPX      ;IF NOT: BR
4275 022416 032777 000200 156144      BIT     #200,@SWR  ;SEE IF STATUS CHECK
4276 022424 001002      BNE     EORP3      ;IF NOT: BR
4277 022426 004767 175130      JSR     PC,WSTCK   ;ELSE GO CHECK STATUS
4278 022432 000240      EORP3:  NOP
4279 022434 032777 000400 156126      BIT     #400,@SWR ;SEE IF DATA CHECK
4280 022442 001002      BNE     EORPX      ;IF NOT: BR
4281 022444 004767 175404      JSR     PC,DCHK    ;ELSE GO CHECK DATA
4282 022450 000240      EORPX:  NOP
4283 022452 005067 156220      CLR     TEMP2      ;CLEAR FLAG
4284 022456 000207      RTS      PC         ;RETURN
4285

```



```
4286                                     ;LOGIC TEST ADDRESSING ERROR SUBROUTINE*****
4287
4288 022460 005067 156232          LTGER3: CLR      EXFL
4289 022464 012767 027431 156200    MOV      #MSG51,ERADD
4290 022472 012767 000001 156246    LTGER0: MOV      #1,ADDFL          ;SET NO ADDRESS FLAG
4291 022500 000240                                LTGER:  NOP
4292 022502 005067 156160                                CLR      PFLG          ;CLEAR PRINT FLAG
4293 022506 032777 020000 156054    BIT      #20000,@SWR    ;SEE IF SHOULD PRINT
4294 022514 001402                                BEQ      LTGA          ;IF SO: BR
4295 022516 000167 000224                                JMP      LTGX          ;ELSE GO TO EXIT
4296 022522 005767 156072          LTGA:  TST      HDRFL    ;SEE IF PRINTED HEADER
4297 022526 001004                                BNE      LTGA1        ;IF SO: BR
4298 022530 016704 156066                                MOV      EMADDR,R4
4299 022534 004767 002024                                JSR      PC,TTOUT
4300 022540 012767 000001 156052    LTGA1: MOV      #1,HDRFL    ;PRINT TEST HEADER
4301 022546 016704 156120                                MOV      ERADD,R4     ;SET HEADER FLAG
4302 022552 004767 002006                                JSR      PC,TTOUT
4303 022556 005767 156164                                TST      ADDFL
4304 022562 001003                                BNE      LTGA2
4305 022564 010103                                MOV      R1,R3
4306 022566 004767 002120                                JSR      PC,OCTP
4307 022572 005767 156120          LTGA2: TST      EXFL          ;PRINT ADDRESS
4308 022576 001412                                BEQ      LTGC          ;IF NO STATUS: BR
4309 022600 012704 025702                                MOV      #MSG6,R4
4310 022604 022767 000001 156104    CMP      #1,EXFL      ;EXPT-NOT RCVD
4311 022612 001402                                BEQ      LTGB
4312 022614 012704 025721                                MOV      #MSG7,R4
4313 022620 004767 001740          LTGB:  JSR      PC,TTOUT    ;RCVD-NOT EXPT
4314 022624 005267 156036          LTGC:  INC      PFLG      ;PRINT STATUS
4315 022630 005767 156112                                TST      ADDFL        ;SEE IF ADD TST
4316 022634 001430                                BEQ      LTGD          ;IF SO: BR
4317 022636 005767 156102                                TST      T24FL        ;SEE IF TEST 24
4318 022642 001423                                BEQ      LTGCO        ;IF NOT: BR
4319 022644 012704 035264                                MOV      #WMSG27,R4
4320 022650 004767 001710                                JSR      PC,TTOUT
4321 022654 012704 026200                                MOV      #MSG12,R4
4322 022660 004767 001700                                JSR      PC,TTOUT
4323 022664 012703 177777                                MOV      #-1,R3
4324 022670 004767 002006                                JSR      PC,OCTPE
4325 022674 012704 026207                                MOV      #MSG13,R4
4326 022700 004767 001660                                JSR      PC,TTOUT
4327 022704 010103                                MOV      R1,R3
4328 022706 004767 001770                                JSR      PC,OCTPE
4329 022712 004767 000100          LTGCO: JSR      PC,REGP    ;PRINT RCVD
4330 022716 032777 010000 155644    LTGD:  BIT      #10000,@SWR ;PRINT RCVD
4331 022724 001010                                BNE      LTGX
4332 022726 012704 026277                                MOV      #MSG16,R4
4333 022732 004767 001626                                JSR      PC,TTOUT
4334 022736 016703 155740                                MOV      ITCNT,R3
4335 022742 004767 001744                                JSR      PC,OCTP
4336 022746 005777 155616          LTGX:  TST      @SWR
4337 022752 100001                                BPL      LTGXA
4338 022754 000000                                HALT
4339 022756 005767 155704          LTGXA: TST      PFLG
4340 022762 001006                                BNE      LTGXX
4341 022764 032777 020000 155576    BIT      #20000,@SWR ;IF PRINTED: BR
```

```
4342 022772 001002          BNE      LTGXX      ;IF STILL NO PRINT: BR
4343 022774 000167 177522    JMP      LTGA       ;ELSE GO PRINT ERROR
4344 023000 005067 155742    LTGXX:  CLR      ADDFL ;CLEAR ADDRESS FLAG
4345 023004 005067 155706    CLR      EXFL
4346 023010 000167 001056    JMP      SCOPE
4347 023014 000207          RTS      PC         ;EXIT
4348
4349                          ;SUBROUTINE TO PRINT MAJOR REGISTERS*****
4350
4351 023016 000240          REGP:  NOP
4352 023020 012704 027254    MOV      #MSG46,R4
4353 023024 004767 001534    JSR      PC,TTOUT   ;PRINT REGISTER HEADER
4354 023030 017703 155454    MOV      @C1,R3
4355 023034 004767 001642    JSR      PC,OCTPE
4356 023040 017703 155446    MOV      @WC,R3
4357 023044 004767 001632    JSR      PC,OCTPE
4358 023050 017703 155440    MOV      @BA,R3
4359 023054 004767 001622    JSR      PC,OCTPE
4360 023060 017703 155432    MOV      @FC,R3
4361 023064 004767 001612    JSR      PC,OCTPE
4362 023070 017703 155424    MOV      @CS,R3
4363 023074 004767 001602    JSR      PC,OCTPE
4364 023100 017703 155416    MOV      @DS,R3
4365 023104 004767 001572    JSR      PC,OCTPE   ;PRINT REGISTERS
4366 023110 017703 155410    MOV      @ER,R3
4367 023114 004767 001562    JSR      PC,OCTPE
4368 023120 017703 155402    MOV      @AS,R3
4369 023124 004767 001552    JSR      PC,OCTPE
4370 023130 017703 155400    MOV      @MR,R3
4371 023134 004767 001542    JSR      PC,OCTPE
4372 023140 017703 155376    MOV      @TC,R3
4373 023144 004767 001532    JSR      PC,OCTPE
4374 023150 000207          RTS      PC
4375
4376
```



```

4377                                     ;DRIVE CLEAR SUBROUTINE*****
4378
4379 023152 000240                       DRVCLR: NOP
4380 023154 012704 040000                MOV     #40000,R4
4381 023160 005304                       DCD:    DEC     R4
4382 023162 001376                       BNE    DCD          ;DELAY
4383 023164 005067 155476                CLR    PFLG
4384 023170 004767 000224                JSR    PC,ATTN     ;GO SEE OF ATTN SET
4385 023174 012777 000011 155306        MOV    #11,@C1    ;ISSUE DRIVE CLEAR
4386 023202 005000                       CLR    R0
4387 023204 032777 000200 155310        DCA:   BIT    #200,@DS ;SEE IF DRY
4388 023212 001002                       BNE    DCA0
4389 023214 005300                       DEC    R0
4390 023216 001372                       BNE    DCA          ;WAIT FOR DRY
4391 023220 032777 040000 155274        DCA0:  BIT    #40000,@DS ;SEE IF ERR RESET
4392 023226 001024                       BNE    DCE          ;IF NOT: BR
4393 023230 005777 155270                TST   @ER          ;SEE IF ERROR REGISTER RESET
4394 023234 001021                       BNE    DCE          ;IF NOT: BR
4395 023236 005777 155260                TST   @DS          ;SEE IF ATA RESET
4396 023242 100416                       BMI    DCE          ;IF NOT: BR
4397 023244 012703 000001                MOV    #1,R3      ;SET TEST BIT
4398 023250 016704 155350                MOV    DRVN,R4    ;GET DRIVE NUMBER
4399 023254 005704                       TST   R4          ;SEE IF DRIVE 0
4400 023256 001404                       BEQ   DCC          ;IF SO: BR
4401 023260 000241                       DCB:   CLC
4402 023262 006103                       ROL    R3          ;POSITION TEST BIT PER DRIVE NUMBER
4403 023264 005304                       DEC    R4          ;SEE IF DONE
4404 023266 001374                       BNE    DCB          ;IF NOT: BR
4405 023270 030377 155232                DCC:   BIT    R3,@AS ;SEE IF ATTEN IS RESET
4406 023274 001001                       BNE    DCE          ;IF NOT: BR
4407 023276 000207                       RTS    PC          ;RETURN
4408 023300 000240                       DCE:   NOP
4409 023302 032777 020000 155260        BIT    #20000,@SWR ;SEE IF ERROR PRINT INHIBIT
4410 023310 001017                       BNE    DCEX        ;IF SO: BR
4411 023312 005767 155302                TST   HDRFL       ;SEE IF PRINT HEADER
4412 023316 001004                       BNE    DCEA        ;IF NOT: BR
4413 023320 016704 155276                MOV    EMADDR,R4
4414 023324 004767 001234                JSR    PC,TTOUT    ;PRINT HEADER
4415 023330 012704 027360                DCEA:  MOV    #MSG47,R4
4416 023334 004767 001224                JSR    PC,TTOUT    ;PRINT DRIVE CLEAR ERROR
4417 023340 004767 177452                JSR    PC,REGP     ;PRINT REGISTERS
4418 023344 005267 155316                INC    PFLG        ;SET PRINTED FLAG
4419 023350 005777 155214                DCEX:  TST   @SWR   ;SEE IF HALT ON ERROR
4420 023354 100001                       BPL   DCEXA       ;IF NOT: BR
4421 023356 000000                       HALT
4422 023360 005767 155302                DCEXA: TST   PFLG   ;SEE IF HAVE PRINTED
4423 023364 001006                       BNE    DCEXX       ;IF SO: BR
4424 023366 032777 020000 155174        BIT    #20000,@SWR ;SEE IF SHOULD PRINT
4425 023374 001002                       BNE    DCEXX       ;IF NOT: BR
4426 023376 000167 177676                JMP   DCE          ;ELSE PRINT THIS ERROR
4427 023402 000240                       DCEXX: NOP
4428 023404 012767 023152 155300        MOV    #DRVCLR,SCOLP ;SET SCOPE LOOP ADDRESS
4429 023412 000167 000454                JMP   SCOPE        ;GO DO SCOPE LOOP
4430 023416 000207                       RTS    PC          ;RETURN

```



```

4431                                     ;COMPOSITE ERROR CHECK SUBROUTINE*****
4432
4433 023420 000240                      ATTN:  NOP
4434 023422 005777 155074                TST    @DS          ;SEE IF ATA SET
4435 023426 001005                      BNE    ATTA        ;IF SO: BR
4436 023430 012767 026716 155242        MOV    #MSG32,TEMP3
4437 023436 000167 000064                JMP    ATTP        ;ELSE PRINT ERROR
4438 023442 032777 040000 155052        ATTA:  BIT    #40000,@DS ;SEE IF COMPOSITE ERROR SET
4439 023450 001005                      BNE    ATTB        ;IF SO: BR
4440 023452 012767 026700 155220        MOV    #MSG31,TEMP3
4441 023460 000167 000042                JMP    ATTP        ;ELSE PRINT ERROR
4442 023464 012703 000001                ATTB:  MOV    #1,R3  ;SET TEST BIT
4443 023470 012767 026734 155202        MOV    #MSG33,TEMP3
4444 023476 016704 155122                MOV    DRVN,R4     ;GET DRIVE NUMBER
4445 023502 005704                      TST    R4          ;SEE IF DRIVE 0
4446 023504 001404                      BEQ    ATTD        ;IF SO: BR
4447 023506 000241                ATTC:  CLC
4448 023510 006103                      ROL    R3          ;POSITION TEST BIT
4449 023512 005304                      DEC    R4          ;SEE IF DONE
4450 023514 001374                      BNE    ATTC        ;IF NOT: BR
4451 023516 030377 155004                ATTD:  BIT    R3,@AS ;SEE IF ATTEN SUMMARY SET
4452 023522 001401                      BEQ    ATTP        ;IF NOT: BR
4453 023524 000207                      RTS    PC          ;ELSE RETURN
4454 023526 032777 020000 155034        ATTP:  BIT    #20000,@SWR ;SEE IF PRINT INHIBIT
4455 023534 001021                      BNE    ATTX        ;IF SO: BR
4456 023536 005767 155056                TST    HDRFL      ;SEE IF DONE HEADER
4457 023542 001004                      BNE    ATTPA       ;IF SO: BR
4458 023544 016704 155052                MOV    EMADDR,R4
4459 023550 004767 001010                JSR    PC,TTOUT    ;PRINT HEADER
4460 023554 016704 155120                ATTPA: MOV    TEMP3,R4
4461 023560 004767 001000                JSR    PC,TTOUT    ;PRINT ERROR TYPE
4462 023564 004767 177226                JSR    PC,REGP     ;PRINT REGISTERS
4463 023570 005267 155072                INC    PFLG       ;SET PRINT FLAG
4464 023574 005267 155020                INC    HDRFL      ;SET HEADER FLAG
4465 023600 005777 154764                ATTX:  TST    @SWR  ;SEE IF HALT ON ERROR
4466 023604 100001                      BPL    ATTXA      ;IF NOT: BR
4467 023606 000000                      HALT
4468 023610 005767 155052                ATTXA: TST    PFLG  ;SEE IF DONE PRINT
4469 023614 001006                      BNE    ATTXX      ;IF SO: BR
4470 023616 032777 020000 154744        BIT    #20000,@SWR ;SEE IF SHOULD PRINT
4471 023624 001002                      BNE    ATTXX      ;IF NOT: BR
4472 023626 000167 177674                JMP    ATTP        ;ELSE PRINT ERROR
4473 023632 005067 155030                ATTXX: CLR    PFLG  ;CLEAR PRINT FLAG
4474 023636 000207                      RTS    PC          ;RETURN

```

```

4475                                     ;LOGIC TEST REGISTER BIT ERROR SUBROUTINE*****
4476
4477 023640 012767 000001 155070 LTGER2: MOV #1,PEXFL ;SET FLAG
4478 023646 000240 LTGER1: NOP
4479 023650 005067 155012 CLR PFLG ;CLEAR PRINT FLAG
4480 023654 032777 020000 154706 BIT #20000,@SWR ;SEE IF PRINT ERRORS
4481 023662 001402 BEQ LTG1A ;IF SO: BR
4482 023664 000167 000132 JMP LTG1X ;ELSE GO TO EXIT
4483 023670 005767 154724 LTG1A: TST HDRFL ;SEE IF PRINT HEADER
4484 023674 001004 BNE LTG1B ;IF NOT: BR
4485 023676 016704 154720 MOV EMADDR,R4
4486 023702 004767 000656 JSR PC,TTOUT ;PRINT HEADER
4487 023706 012767 000001 154704 LTG1B: MOV #1,HDRFL ;SET FLAG
4488 023714 016704 154752 MOV ERADD,R4
4489 023720 004767 000640 JSR PC,TTOUT ;PRINT ERROR CODE
4490 023724 005767 155006 TST PEXFL ;SEE IF PRINT EXPT-RCVD
4491 023730 001016 BNE LTG1T ;IF NOT: BR
4492 023732 012704 026200 MOV #MSG12,R4
4493 023736 004767 000622 JSR PC,TTOUT ;PRINT EXPT TAG
4494 023742 010103 MOV R1,R3
4495 023744 004767 000742 JSR PC,OCTP ;PRINT EXPT
4496 023750 012704 026207 MOV #MSG13,R4
4497 023754 004767 000604 JSR PC,TTOUT ;PRINT RCVD TAG
4498 023760 010203 MOV R2,R3
4499 023762 004767 000724 JSR PC,OCTP ;PRINT RCVD
4500 023766 032777 010000 154574 LTG1T: BIT #10000,@SWR
4501 023774 001010 BNE LTG1C
4502 023776 012704 026277 MOV #MSG16,R4
4503 024002 004767 000556 JSR PC,TTOUT
4504 024006 016703 154670 MOV ITCNT,R3
4505 024012 004767 000674 JSR PC,OCTP ;PRINT ITERATION
4506 024016 005267 154644 LTG1C: INC PFLG
4507 024022 000240 LTG1X: NOP
4508 024024 005777 154540 TST @SWR
4509 024030 100001 BPL LTG1X1 ;IF NOT STOP ON ERROR: BR
4510 024032 000000 HALT
4511 024034 005767 154626 LTG1X1: TST PFLG
4512 024040 001006 BNE LTG1XX ;IF HAVE PRINTED: BR
4513 024042 032777 020000 154520 BIT #20000,@SWR
4514 024050 001002 BNE LTG1XX ;IF STILL NO PRINT: BR
4515 024052 000167 177612 JMP LTG1A ;ELSE PRINT ERROR
4516 024056 000240 LTG1XX: NOP
4517 024060 005067 154652 CLR PEXFL ;CLEAR EXPT-RCVD FLAG
4518 024064 000167 000002 JMP SCOPE ;GO TO SCOPE
4519 024070 000207 RTS PC ;RETURN
4520

```



```

4521
4522                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
4523
4524 024072 000240 SCOPE: NOP
4525 024074 032777 040000 154466 BIT #40000,@SWR ;SEE IF LOOP ON ERROR
4526 024102 001001 BNE SCOPE1 ;IF SO: BR
4527 024104 000207 RTS PC ;ELSE EXIT
4528 024106 000240 SCOPE1: NOP
4529 024110 005726 TST (SP)+ ;RESET STACK
4530 024112 000240 NOP
4531 024114 000240 NOP
4532 024116 000177 154570 JMP @SCOLP ;LOOP ON ERROR
4533
4534                                     ;TEST ITERATION SUBROUTINE*****
4535
4536 024122 000240 ITER: NOP
4537 024124 032777 010000 154436 BIT #10000,@SWR ;SEE IF ITERATIONS
4538 024132 001403 BEQ ITER1 ;IF SO: BR
4539 024134 005067 154542 ITER0: CLR ITCNT ;CLEAR ITERATION COUNTER
4540 024140 000207 RTS PC ;ELSE EXIT
4541 024142 005267 154534 ITER1: INC ITCNT ;BUMP COUNTER
4542 024146 026767 154530 154432 CMP ITCNT,ITAMT ;SEE IF DONE ALL
4543 024154 001767 BEQ ITER0 ;IF SO: BR
4544 024156 005726 TST (SP)+ ;RESET STACK
4545 024160 017700 154530 MOV @ITRLP,R0 ;SET ITERATION POINTER
4546 024164 000110 JMP (R0) ;GO ITERATE
4547
4548                                     ;MANUAL INTERVENTION INHIBIT*****
4549
4550 024166 000240 INMT: NOP
4551 024170 012704 027104 MOV #MSG43,R4
4552 024174 004767 000364 JSR PC,TTOUT ;GO PRINT INHIB MSG
4553 024200 000000 HALT
4554 024202 000167 155624 JMP TSCD2 ;RETURN TO SCHED
4555

```



```

4556
4557
4558
4559 024206 000240
4560 024210 012777 000040 154302
4561 024216 016777 154402 154274
4562 024224 016777 154434 154310
4563 024232 000207
4564
4565
4566
4567 024234 000240
4568 024236 004767 000322
4569 024242 012704 033416
4570 024246 004767 000312
4571 024252 012705 000700
4572 024256 012701 000001
4573 024262 012702 177777
4574 024266 012703 000000
4575 024272 004767 000030
4576 024276 000240
4577 024300 000207
4578
4579
4580
4581 024302 000240
4582 024304 022626
4583 024306 000240
4584 024310 000240
4585 024312 000177 154352
4586
4587
4588
4589 024316 000240
4590 024320 000240
4591 024322 000240
4592 024324 000002
4593

```

```

;INITIALIZE SUBROUTINE*****
INIT1: NOP
      MOV #40,@CS ;INIT
INIT2: MOV DRVN,@CS ;SELECT DRIVE
      MOV SLVN,@TC ;SELECT SLAVE
      RTS PC ;RETURN

;MANUAL INSTRUCTION SUBROUTINE*****
INST:  NOP
      JSR PC,TTOUT ;PRINT INSTRUCTION
      MOV #MMSG0,R4
      JSR PC,TTOUT ;PRINT REPLY
      MOV #TEMP3,R5
      MOV #1,R1
      MOV #-1,R2
      MOV #0,R3
      JSR PC,TTR ;AWAIT REPLY
      NOP
      RTS PC ;EXIT

;MAG TAPE INTERRUPT HANDLER*****
MTINT: NOP
      CMP (SP)+,(SP)+ ;RESET STACK POINTER
      NOP
      NOP
      JMP @RTRN ;RETURN TO CALLER

;TTY INTERRUPT HANDLER*****
TTINT: NOP
      NOP
      NOP
      RTI

```

```

4594 ;*****
4595 ;TTY ENTRY SUBROUTINE:
4596 ;
4597 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
4598 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
4599 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
4600 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
4601 ; (0-7) AND MUST FALL WITHIN THE LIMITS SET BY
4602 ;THE CALLING ROUTINE.
4603 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
4604 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
4605 ;MAY BE REENTERED.
4606 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
4607 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
4608 ;CARRIAGE RETURN
4609 ;*****
4610
4611 024326 005067 154342 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
4612 024332 005000 CLR RO
4613 024334 004767 000152 TTR0: JSR PC,TTIN ;GO READ CHARACTER
4614 024340 122767 000215 154250 CMPB #215,TIB ;SEE IF CR
4615 024346 001005 BNE TTR1 ;IF NOT: BR
4616 024350 005767 154320 TST TEMP1 ;SEE IF FIRST CHARACTER
4617 024354 001446 BEQ TTR5 ;IF SO: BR
4618 024356 000167 000066 JMP TTR2 ;ELSE GO LOAD VALUE
4619 024362 122767 000260 154226 TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN 0
4620 024370 101402 BLOS TTR1A ;IF NOT: BR
4621 024372 000167 000076 JMP TINER ;ELSE GO TO ERROR
4622 024376 122767 000270 154212 TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THAN 7
4623 024404 101002 BHI TTR1B ;IF NOT: BR
4624 024406 000167 000062 JMP TINER ;ELSE GO TO ERROR
4625 024412 005267 154256 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
4626 024416 000241 CLC
4627 024420 006100 ROL RO
4628 024422 000241 CLC
4629 024424 006100 ROL RO ;SHIFT 3 LEFT
4630 024426 000241 CLC
4631 024430 006100 ROL RO
4632 024432 042767 177770 154156 BIC #177770,TIB ;STRIP ASCII
4633 024440 056700 154152 BIS TIB,RO ;LOAD CHARACTER
4634 024444 005301 DEC R1 ;SEE IF DONE
4635 024446 001332 BNE TTR0 ;IF NOT: BR
4636 024450 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
4637 024452 101402 BLOS TTR3 ;IF NOT: BR
4638 024454 000167 000014 JMP TINER ;ELSE GO TO ERROR
4639 024460 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT
4640 024462 101402 BLOS TTR4 ;IF NOT: BR
4641 024464 000167 000004 JMP TINER ;ELSE GO TO ERROR
4642 024470 010015 TTR4: MOV RO,(R5) ;LOAD VALUE
4643 024472 000207 TTR5: RTS PC ;EXIT
4644

```



```

4645
4646
4647
4648 024474 012704 027044
4649 024500 004767 000060
4650 024504 162716 000020
4651 024510 000207
4652
4653
4654
4655 024512 005077 154054
4656 024516 005077 154052
4657 024522 005067 154070
4658 024526 005277 154040
4659 024532 105777 154034
4660 024536 100375
4661 024540 017767 154030 154050
4662 024546 105777 154024
4663 024552 100375
4664 024554 116777 154036 154016
4665 024562 000207
4666
4667
4668
4669 024564 112467 154024
4670 024570 122767 000043 154016
4671 024576 001440
4672 024600 122767 000045 154006
4673 024606 001403
4674 024610 004767 000050
4675 024614 000763
4676 024616 112767 000015 153770
4677 024624 004767 000034
4678 024630 012703 000004
4679 024634 005067 153754
4680 024640 004767 000020
4681 024644 005303
4682 024646 001372
4683 024650 112767 000012 153736
4684 024656 004767 000002
4685 024662 000740
4686 024664 105777 153706
4687 024670 100375
4688 024672 116777 153716 153700
4689 024700 000207
4690
4691
4692
4693
4694 024702 012767 000001 000226
4695 024710 000402
4696 024712 005067 000220
4697 024716 010304
4698 024720 001007
4699 024722 005767 000210
4700 024726 001004

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

;TTY READ SUBROUTINE*****
TTIN: CLR @TKS
       CLR @TKB
       CLR TIB
       INC @TKS
TTIN1: TSTB @TKS
       BPL TTIN1
       MOV @TKB,TIB
TTIN2: TSTB @TPS
       BPL TTIN2
       MOVB TIB,@TPB
       RTS PC

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

;OCTAL OUTPUT SUBROUTINE*****
OCTPE: MOV #1,OFL
       BR OCTPE1
OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
        BNE OCTP0 ;IF NOT ZERO: BR
        TST OFL ;SEE IF PRINT ALL 0
        BNE OCTP0 ;IF SO: BR

```



4701	024730	004767	000162		JSR	PC,OCTPG1	:ELSE PRINT ZERO
4702	024734	000167	000120		JMP	OCTP3	:SPACE AND EXIT
4703	024740	032704	100000	OCTP0:	BIT	#100000,R4	:SEE IF MSD = 1
4704	024744	001406			BEQ	OCTP1	:IF NOT: BR
4705	024746	012704	000001		MOV	#1,R4	
4706	024752	004767	000116		JSR	PC,OCTPG	:PRINT 1
4707	024756	000167	000006		JMP	OCTP2	
4708	024762	005004		OCTP1:	CLR	R4	
4709	024764	004767	000104		JSR	PC,OCTPG	:PRINT 0
4710	024770	010304		OCTP2:	MOV	R3,R4	
4711	024772	006004			ROR	R4	
4712	024774	006004			ROR	R4	
4713	024776	006004			ROR	R4	:POSITION DIGIT
4714	025000	006004			ROR	R4	
4715	025002	000304			SWAB	R4	
4716	025004	004767	000064		JSR	PC,OCTPG	:PRINT DIGIT 2
4717	025010	010304			MOV	R3,R4	
4718	025012	006004			ROR	R4	
4719	025014	000304			SWAB	R4	
4720	025016	004767	000052		JSR	PC,OCTPG	:PRINT DIGIT 3
4721	025022	010304			MOV	R3,R4	
4722	025024	006104			ROL	R4	
4723	025026	006104			ROL	R4	
4724	025030	000304			SWAB	R4	
4725	025032	004767	000036		JSR	PC,OCTPG	:PRINT DIGIT 4
4726	025036	010304			MOV	R3,R4	
4727	025040	006004			ROR	R4	
4728	025042	006004			ROR	R4	
4729	025044	006004			ROR	R4	
4730	025046	004767	000022		JSR	PC,OCTPG	
4731	025052	010304			MOV	R3,R4	
4732	025054	004767	000014		JSR	PC,OCTPG	:PRINT DIGIT 5
4733	025060	012767	000240	153526	OCTP3:	MOV	#240,TOB
4734	025066	004767	177572		JSR	PC,TOG	:PRINT SPACE
4735	025072	000207			RTS	PC	:EXIT
4736	025074	042704	177770		OCTPG:	BIC	#177770,R4
4737	025100	001004			BNE	OCTPG0	
4738	025102	005767	000030		TST	OFL	
4739	025106	001001			BNE	OCTPG0	
4740	025110	000207			RTS	PC	
4741	025112	005267	000020		OCTPG0:	INC	OFL
4742	025116	052704	000260		OCTPG1:	BIS	#260,R4
4743	025122	010467	153466		MOV	R4,TOB	
4744	025126	004767	177532		JSR	PC,TOG	
4745	025132	010304			MOV	R3,R4	
4746	025134	000207			RTS	PC	
4747	025136	000000		OFL:	0		:FIRST CHAR FLAG
4748							

```

4749
4750
4751
4752 025140 005067 153450
4753 025144 012704 000010
4754 025150 110367 153440
4755 025154 105777 153416
4756 025160 100375
4757 025162 132767 000200 153424
4758 025170 001404
4759 025172 012777 000061 153400
4760 025200 000403
4761 025202 012777 000060 153370
4762 025210 006167 153400
4763 025214 005304
4764 025216 001356
4765 025220 000207
4766 025222 016703 153452
4767 025226 000303
4768 025230 004767 177704
4769 025234 016703 153440
4770 025240 004767 177674
4771 025244 000207
4772
4773
4774
4775 025246 010304
4776 025250 000304
4777 025252 006004
4778 025254 006004
4779 025256 006004
4780 025260 006004
4781 025262 004767 000036
4782 025266 010304
4783 025270 000304
4784 025272 004767 000026
4785 025276 010304
4786 025300 006004
4787 025302 006004
4788 025304 006004
4789 025306 006004
4790 025310 004767 000010
4791 025314 010304
4792 025316 004767 000002
4793 025322 000207
4794 025324 012767 000260 153262
4795 025332 042704 177760
4796 025336 050467 153252
4797 025342 004767 177316
4798 025346 000207

;DATA CHARACTER OUTPUT SUBROUTINE*****
DOUT: CLR TOB
MOV #10,R4 ;SET NUMBER TO PRINT
MOV B3,TOB
DOUT1: TSTB @TPS
BPL DOUT1
BITB #200,TOB
BEQ DOUT2
MOV #061,@TPB
BR DOUT3
DOUT2: MOV #060,@TPB
DOUT3: ROL TOB
DEC R4
BNE DOUT1
RTS PC
DOUTD: MOV TEMP3,R3
SWAB R3
JSR PC,DOUT
MOV TEMP3,R3
JSR PC,DOUT
RTS PC

;TU45 SERIAL NUMBER PRINT SUBROUTINE*****
SNPT: MOV R3,R4
SWAB R4
ROR R4
ROR R4
ROR R4
ROR R4
ROR R4 ;GET FIRST DIGIT
JSR PC,SNPG ;PRINT
MOV R3,R4
SWAB R4 ;GET SECOND DIGIT
JSR PC,SNPG ;PRINT
MOV R3,R4
ROR R4
ROR R4
ROR R4
ROR R4
JSR PC,SNPG ;PRINT THIRD DIGIT
MOV R3,R4
JSR PC,SNPG ;PRINT FOURTH DIGIT
RTS PC ;EXIT
SNPG: MOV #260,TOB ;SET BASE = 0
BIC #177760,R4 ;MASK DIGIT
BIS R4,TOB ;SET ASCII
JSR PC,TOG ;TYPE DIGIT
RTS PC ;RETURN
    
```



```

4799                                     :MESSAGE TABLE*****
4800
4801 025350 022445 046524 031060 MSG1: .ASCII /%TM02-TU45 CONTROL LOGIC TEST (CZTUKA0)#/
4802 025356 052055 032125 020065
4803 025364 047503 052116 047522
4804 025372 020114 047514 044507
4805 025400 020103 042524 052123
4806 025406 024040 055103 052524
4807 025414 040513 024460 043
4808 025421 105 052116 051105 .ASCII /ENTER CONDITIONS IN OCTALX#/
4809 025426 041440 047117 044504
4810 025434 044524 047117 020123
4811 025442 047111 047440 052103
4812 025450 046101 021445
4813 025454 042045 044522 042526 MSG2: .ASCII /%DRIVE NUMBER #/
4814 025462 047040 046525 042502
4815 025470 020122 043
4816 025473 045 043045 051117 MSG2A: .ASCII /%%FOR DRIVE ADDRESS TEST;/
4817 025500 042040 044522 042526
4818 025506 040440 042104 042522
4819 025514 051523 052040 051505
4820 025522 035524
4821 025524 020045 047105 042524 .ASCII /% ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
4822 025532 020122 054105 052120
4823 025540 042040 044522 042526
4824 025546 047040 046525 042502
4825 025554 026122 040440 046114
4826 025562 047440 044124 051105
4827 025570 020123 044123 052517
4828 025576 042114 041040 020105
4829 025604 047516 026516 054105
4830 025612 051511 040524 052116
4831 025620 021456
4832 025622 047045 047117 042455 MSG3: .ASCII /%NON-EXIST DRIVE #/
4833 025630 044530 052123 042040
4834 025636 044522 042526 021440
4835 025644 051045 020110 042504 MSG4: .ASCII /%RH DETECTED #/
4836 025652 042524 052103 042105
4837 025660 021440
4838 025662 052045 030115 020062 MSG5: .ASCII /%TM02 DETECTED #/
4839 025670 042504 042524 052103
4840 025676 042105 021440
4841 025702 054105 052120 047055 MSG6: .ASCII /EXPT-NOT RECVD#/
4842 025710 052117 051040 041505
4843 025716 042126 043
4844 025721 122 053103 026504 MSG7: .ASCII /RCVD-NOT EXPT#/
4845 025726 047516 020124 054105
4846 025734 052120 043
4847 025737 045 046123 053101 MSG8: .ASCII /%SLAVE NUMBER #/
4848 025744 020105 052516 041115
4849 025752 051105 021440
4850 025756 022445 047506 020122 MSG8A: .ASCII /%%FOR SLAVE ADDRESS TEST;/
4851 025764 046123 053101 020105
4852 025772 042101 051104 051505
4853 026000 020123 042524 052123
4854 026006 073

```



4855	026007	045	042440	052116					
4856	026014	051105	042440	050130					
4857	026022	020124	046123	053101					
4858	026030	020105	052516	041115					
4859	026036	051105	020054	046101					
4860	026044	020114	052117	042510					
4861	026052	051522	051440	047510					
4862	026060	046125	020104	042502					
4863	026066	047040	047117	042455					
4864	026074	044530	052123	047101					
4865	026102	027124	043						
4866	026105	045	047516	026516	MSG9:	.ASCII	/%NON-EXIST SLAVE #/		
4867	026112	054105	051511	020124					
4868	026120	046123	053101	020105					
4869	026126	043							
4870	026127	045	042522	042101	MSG10:	.ASCII	/%READ CONT BUS PAR #/		
4871	026134	041440	047117	020124					
4872	026142	052502	020123	040520					
4873	026150	020122	043						
4874	026153	045	051127	052111	MSG11:	.ASCII	/%WRITE CONT BUS PAR #/		
4875	026160	020105	047503	052116					
4876	026166	041040	051525	050040					
4877	026174	051101	021440						
4878	026200	042440	050130	020124	MSG12:	.ASCII	/ EXPT #/		
4879	026206	043							
4880	026207	040	041522	042126	MSG13:	.ASCII	/ RCVD #/		
4881	026214	021440							
4882	026216	046445	020122	044502	MSG14:	.ASCII	/%MR BITS 4-0#/		
4883	026224	051524	032040	030055					
4884	026232	043							
4885	026233	045	051115	041040	MSG15:	.ASCII	/%MR BITS 15-7#/		
4886	026240	052111	020123	032461					
4887	026246	033455	043						
4888	026251	045	051115	041040	MSG15A:	.ASCII	/%MR BIT6 DID NOT SET #/		
4889	026256	052111	020066	044504					
4890	026264	020104	047516	020124					
4891	026272	042523	020124	043					
4892	026277	045	052111	051105	MSG16:	.ASCII	/%ITER: #/		
4893	026304	020072	043						
4894	026307	045	041524	041040	MSG17:	.ASCII	/%TC BIT 14 #/		
4895	026314	052111	030440	020064					
4896	026322	043							
4897	026323	045	041524	041040	MSG18:	.ASCII	/%TC BITS 12-0 #/		
4898	026330	052111	020123	031061					
4899	026336	030055	021440						
4900	026342	043045	020103	044502	MSG19:	.ASCII	/%FC BITS 15-0 #/		
4901	026350	051524	030440	026465					
4902	026356	020060	043						
4903	026361	045	052506	020116	MSG20:	.ASCII	/%FUN CODE BITS 5-1 OF C1 #/		
4904	026366	047503	042504	041040					
4905	026374	052111	020123	026465					
4906	026402	020061	043117	041440					
4907	026410	020061	043						
4908	026413	045	047507	041040	MSG21:	.ASCII	/%GO BIT NOT CORRECT AT START #/		
4909	026420	052111	047040	052117					
4910	026426	041440	051117	042522					

4911	026434	052103	040440	020124		
4912	026442	052123	051101	020124		
4913	026450	043				
4914	026451	045	047507	041040	MSG22:	.ASCII /%GO BIT NOT SET #/
4915	026456	052111	047040	052117		
4916	026464	051440	052105	021440		
4917	026472	043445	020117	044502	MSG23:	.ASCII /%GO BIT NOT RESET BY INIT #/
4918	026500	020124	047516	020124		
4919	026506	042522	042523	020124		
4920	026514	054502	044440	044516		
4921	026522	020124	043			
4922	026525	045	051104	020131	MSG24:	.ASCII /%DRY NOT SET BY INIT #/
4923	026532	047516	020124	042523		
4924	026540	020124	054502	044440		
4925	026546	044516	020124	043		
4926	026553	045	051104	020131	MSG25:	.ASCII /%DRY NOT RESET BY GO=1#/
4927	026560	047516	020124	042522		
4928	026566	042523	020124	054502		
4929	026574	043440	036517	021461		
4930	026602	042045	054522	047040	MSG25A:	.ASCII /%DRY NOT SET BY GO=0#/
4931	026610	052117	051440	052105		
4932	026616	041040	020131	047507		
4933	026624	030075	043			
4934	026627	045	047516	044440	MSG26:	.ASCII /%NO INTERRUPT RETURNED#/
4935	026634	052116	051105	052522		
4936	026642	052120	051040	052105		
4937	026650	051125	042516	021504		
4938	026656	041045	042101	051440	MSG27:	.ASCII /%BAD STATUS#/
4939	026664	040524	052524	021523		
4940	026672	051440	035116	021440	MSG30:	.ASCII / SN: #/
4941	026700	042445	051122	047040	MSG31:	.ASCII /%ERR NOT SET #/
4942	026706	052117	051440	052105		
4943	026714	021440				
4944	026716	040445	040524	047040	MSG32:	.ASCII /%ATA NOT SET #/
4945	026724	052117	051440	052105		
4946	026732	021440				
4947	026734	040445	020123	044502	MSG33:	.ASCII /%AS BIT NOT SET #/
4948	026742	020124	047516	020124		
4949	026750	042523	020124	043		
4950	026755	045	041523	047040	MSG34:	.ASCII /%SC NOT SET #/
4951	026762	052117	051440	052105		
4952	026770	021440				
4953	026772	052045	042522	047040	MSG35:	.ASCII /%TRE NOT SET #/
4954	027000	052117	051440	052105		
4955	027006	021440				
4956	027010	051445	040514	047040	MSG36:	.ASCII /%SLA NOT SET #/
4957	027016	052117	051440	052105		
4958	027024	021440				
4959	027026	051445	041523	047040	MSG37:	.ASCII /%SSC NOT SET #/
4960	027034	052117	051440	052105		
4961	027042	021440				
4962	027044	037440	021440		MSG40:	.ASCII / ? #/
4963	027050	022445	047105	020104	MSG41:	.ASCII /%END OF PASS #/
4964	027056	043117	050040	051501		
4965	027064	020123	043			
4966	027067	045	042504	042101	MSG42:	.ASCII /%DEAD TRACK #/



4967	027074	052040	040522	045503	
4968	027102	021440			
4969	027104	022445	040515	052516	MSG43: .ASCII /%MANUAL TESTS (14-17) INHIBITED: HALT%/
4970	027112	046101	052040	051505	
4971	027120	051524	024040	032061	
4972	027126	030455	024467	044440	
4973	027134	044116	041111	052111	
4974	027142	042105	020072	040510	
4975	027150	052114	045		
4976	027153	122	051505	046105	.ASCII /RESELECT AND PRESS CONTINUE%/
4977	027160	041505	020124	047101	
4978	027166	020104	051120	051505	
4979	027174	020123	047503	052116	
4980	027202	047111	042525	021445	
4981	027210	051045	043505	051511	MSG44: .ASCII /%REGISTER START: #/
4982	027216	042524	020122	052123	
4983	027224	051101	035124	021440	
4984	027232	053045	041505	047524	MSG45: .ASCII /%VECTOR ADDRESS: #/
4985	027240	020122	042101	051104	
4986	027246	051505	035123	021440	
4987	027254	041445	030523	020040	MSG46: .ASCII /%CS1 WC BA FC CS2 DS ER AS/
4988	027262	020040	041527	020040	
4989	027270	020040	041040	020101	
4990	027276	020040	020040	041506	
4991	027304	020040	020040	041440	
4992	027312	031123	020040	020040	
4993	027320	051504	020040	020040	
4994	027326	042440	020122	020040	
4995	027334	020040	051501		
4996	027340	020040	020040	046440	.ASCII / MR TC%/
4997	027346	020122	020040	020040	
4998	027354	041524	021445		
4999	027360	047045	052117	051040	MSG47: .ASCII /%NOT RESET BY DRIVE CLEAR%/
5000	027366	051505	052105	041040	
5001	027374	020131	051104	053111	
5002	027402	020105	046103	040505	
5003	027410	021522			
5004	027412	040445	050114	040510	MSG50: .ASCII /%ALPHA NOT SET%/
5005	027420	047040	052117	051440	
5006	027426	052105	043		
5007	027431	045	047125	054105	MSG51: .ASCII /%UNEXPECTED ERROR BITS%/
5008	027436	042520	052103	042105	
5009	027444	042440	051122	051117	
5010	027452	041040	052111	021523	
5011	027460	047045	055122	047440	MSG52: .ASCII /%NRZ ONLY: #/
5012	027466	046116	035131	021440	
5013	027474	041045	042101	046040	MSG53: .ASCII /%BAD LRC #/
5014	027502	041522	021440		
5015	027506	041045	042101	041440	MSG54: .ASCII /%BAD CK #/
5016	027514	020113	043		
5017	027517	045	042523	052524	MSG55: .ASCII /%SETUP ERROR: CHECK WRAP C WITH TEST 50%/
5018	027524	020120	051105	047522	
5019	027532	035122	041440	042510	
5020	027540	045503	053440	040522	
5021	027546	020120	020060	044527	
5022	027554	044124	052040	051505	



TMO2/TU45 CONTROL LOGIC TEST  
CZTUKA.P11 12-JUN-78 16:17

MACY11 30(1046) 22-JUN-78 08:58 PAGE 134

SEQ 0134

5023	027562	020124	030065	043	
5024	027567	045	052123	052101	MSG56: .ASCII /%STATIC TESTS ONLY: #/
5025	027574	041511	052040	051505	
5026	027602	051524	047440	046116	
5027	027610	035131	021440		

```
5028 ;TEST HEADER*****
5029
5030 027614 022445 047514 044507 MSLT1: .ASCII /%LOGIC TEST 1: DRIVE ADDRESSING (M8909 RH)#/
5031 027622 020103 042524 052123
5032 027630 030440 020072 051104
5033 027636 053111 020105 042101
5034 027644 051104 051505 044523
5035 027652 043516 024040 034115
5036 027660 030071 020071 044122
5037 027666 021451
5038 027670 022445 047514 044507 MSLT2: .ASCII /%LOGIC TEST 2: REGISTER ADDRESSING (M8909 RH)#/
5039 027676 020103 042524 052123
5040 027704 031040 020072 042522
5041 027712 044507 052123 051105
5042 027720 040440 042104 042522
5043 027726 051523 047111 020107
5044 027734 046450 034470 034460
5045 027742 051040 024510 043
5046 027747 045 046045 043517 MSLT3: .ASCII /%LOGIC TEST 3: CONTROL BUS TEST (RH M8905 M8909)#/
5047 027754 041511 052040 051505
5048 027762 020124 035063 041440
5049 027770 047117 051124 046117
5050 027776 041040 051525 052040
5051 030004 051505 020124 051050
5052 030012 020110 034115 030071
5053 030020 020065 034115 030071
5054 030026 024471 043
5055 030031 045 046045 043517 MSLT4: .ASCII /%LOGIC TEST 4: SLAVE ADDRESSING (M8905 M8903)#/
5056 030036 041511 052040 051505
5057 030044 020124 035064 051440
5058 030052 040514 042526 040440
5059 030060 042104 042522 051523
5060 030066 047111 020107 046450
5061 030074 034470 032460 046440
5062 030102 034470 031460 021451
5063 030110 022445 047514 044507 MSLT5: .ASCII /%LOGIC TEST 5: MR BIT TEST (M8905)#/
5064 030116 020103 042524 052123
5065 030124 032440 020072 051115
5066 030132 041040 052111 052040
5067 030140 051505 020124 046450
5068 030146 034470 032460 021451
5069 030154 022445 047514 044507 MSLT6: .ASCII /%LOGIC TEST 6: TC BIT TEST (M8905)#/
5070 030162 020103 042524 052123
5071 030170 033040 020072 041524
5072 030176 041040 052111 052040
5073 030204 051505 020124 046450
5074 030212 034470 032460 021451
5075 030220 022445 047514 044507 MSLT7: .ASCII /%LOGIC TEST 7: FC BIT TEST (M8905)#/
5076 030226 020103 042524 052123
5077 030234 033440 020072 041506
5078 030242 041040 052111 052040
5079 030250 051505 020124 046450
5080 030256 034470 032460 021451
5081 030264 022445 047514 044507 MSLT10: .ASCII /%LOGIC TEST 10: FUNCTION BIT TEST (M8905)#/
5082 030272 020103 042524 052123
5083 030300 030440 035060 043040
```



TM02/TU45 CONTROL LOGIC TEST  
CZTUKA.P11 12-JUN-78 16:17

MACY11 30(1046) 22-JUN-78 08:58 PAGE 136

SEQ 0136

5084	030306	047125	052103	047511	
5085	030314	020116	044502	020124	
5086	030322	042524	052123	024040	
5087	030330	034115	030071	024465	
5088	030336	043			
5089	030337	045	046045	043517	MSLT11: .ASCII /%LOGIC TEST 11: GO BIT TEST (M8909)#/
5090	030344	041511	052040	051505	
5091	030352	020124	030461	020072	
5092	030360	047507	041040	052111	
5093	030366	052040	051505	020124	
5094	030374	046450	034470	034460	
5095	030402	021451			
5096	030404	022445	047514	044507	MSLT12: .ASCII /%LOGIC TEST 12: DRIVE READY BIT (M8909)#/
5097	030412	020103	042524	052123	
5098	030420	030440	035062	042040	
5099	030426	044522	042526	051040	
5100	030434	040505	054504	041040	
5101	030442	052111	024040	034115	
5102	030450	030071	024471	043	
5103	030455	045	046045	043517	MSLT13: .ASCII /%LOGIC TEST 13: INTERRUPT TEST (RH)#/
5104	030462	041511	052040	051505	
5105	030470	020124	031461	020072	
5106	030476	047111	042524	051122	
5107	030504	050125	020124	042524	
5108	030512	052123	024040	044122	
5109	030520	021451			
5110	030522	022445	047514	044507	MSLT14: .ASCII /%LOGIC TEST 14: MANUAL STATUS TEST 1#/
5111	030530	020103	042524	052123	
5112	030536	030440	035064	046440	
5113	030544	047101	040525	020114	
5114	030552	052123	052101	051525	
5115	030560	052040	051505	020124	
5116	030566	021461			
5117	030570	022445	047514	044507	MSLT15: .ASCII /%LOGIC TEST 15: MANUAL STATUS TEST 2#/
5118	030576	020103	042524	052123	
5119	030604	030440	035065	046440	
5120	030612	047101	040525	020114	
5121	030620	052123	052101	051525	
5122	030626	052040	051505	020124	
5123	030634	021462			
5124	030636	022445	047514	044507	MSLT16: .ASCII /%LOGIC TEST 16: MANUAL STATUS TEST 3#/
5125	030644	020103	042524	052123	
5126	030652	030440	035066	046440	
5127	030660	047101	040525	020114	
5128	030666	052123	052101	051525	
5129	030674	052040	051505	020124	
5130	030702	021463			
5131	030704	022445	047514	044507	MSLT17: .ASCII /%LOGIC TEST 17: MANUAL STATUS TEST 4#/
5132	030712	020103	042524	052123	
5133	030720	030440	035067	046440	
5134	030726	047101	040525	020114	
5135	030734	052123	052101	051525	
5136	030742	052040	051505	020124	
5137	030750	021464			
5138	030752	022445	047514	044507	MSLT20: .ASCII /%LOGIC TEST 20: ILLEGAL FUNCTION TEST (M8909)#/
5139	030760	020103	042524	052123	



5140	030766	031040	035060	044440	
5141	030774	046114	043505	046101	
5142	031002	043040	047125	052103	
5143	031010	047511	020116	042524	
5144	031016	052123	024040	034115	
5145	031024	030071	024471	043	
5146	031031	045	046045	043517	MSLT21: .ASCII /%%LOGIC TEST 21: RMR(M8909)#/
5147	031036	041511	052040	051505	
5148	031044	020124	030462	020072	
5149	031052	046522	024122	034115	
5150	031060	030071	024471	043	
5151	031065	045	045045	043517	MSLT22: .ASCII /%%LOGIC TEST 22: CPAR(M8909)#/
5152	031072	041511	052040	051505	
5153	031100	020124	031062	020072	
5154	031106	050103	051101	046450	
5155	031114	034470	034460	021451	
5156	031122	022445	047514	044507	MSLT23: .ASCII /%%LOGIC TEST 23: FMT(M8905 M8906)#/
5157	031130	020103	042524	052123	
5158	031136	031040	035063	043040	
5159	031144	052115	046450	034470	
5160	031152	032460	046440	034470	
5161	031160	033060	021451		
5162	031164	022445	047514	044507	MSLT24: .ASCII /%%LOGIC TEST 24: DPAR(M8906 RH)#/
5163	031172	020103	042524	052123	
5164	031200	031040	035064	042040	
5165	031206	040520	024122	034115	
5166	031214	030071	020066	044122	
5167	031222	021451			
5168	031224	022445	047514	044507	MSLT25: .ASCII /%%LOGIC TEST 25: NEF(M8909)#/
5169	031232	020103	042524	052123	
5170	031240	031040	035065	047040	
5171	031246	043105	046450	034470	
5172	031254	034460	021451		
5173	031260	022445	047514	044507	MSLT26: .ASCII /%%LOGIC TEST 26: FCE(M8909)#/
5174	031266	020103	042524	052123	
5175	031274	031040	035066	043040	
5176	031302	042503	046450	034470	
5177	031310	034460	021451		
5178	031314	022445	047514	044507	MSLT27: .ASCII /%%LOGIC TEST 27: ILR(M8909)#/
5179	031322	020103	042524	052123	
5180	031330	031040	035067	044440	
5181	031336	051114	046450	034470	
5182	031344	034460	021451		
5183	031350	022445	047514	044507	MSLT30: .ASCII /%%LOGIC TEST 30:DTE(M8906 RH)#/
5184	031356	020103	042524	052123	
5185	031364	031440	035060	052104	
5186	031372	024105	034115	030071	
5187	031400	020066	044122	021451	
5188	031406	022445	047514	044507	MSLT31: .ASCII /%%LOGIC TEST 31: OPI(M8903)#/
5189	031414	020103	042524	052123	
5190	031422	031440	035061	047440	
5191	031430	044520	046450	034470	
5192	031436	031460	021451		
5193	031442	022445	047514	044507	MSLT32: .ASCII /%%LOGIC TEST 32: UNS(M8909)#/
5194	031450	020103	042524	052123	
5195	031456	031440	035062	052440	

5196	031464	051516	046450	034470	
5197	031472	034460	021451		
5198	031476	022445	047514	044507	MSLT33: .ASCII /%%LOGIC TEST 33: PIP(M8909)#/
5199	031504	020103	042524	052123	
5200	031512	031440	035063	050040	
5201	031520	050111	046450	034470	
5202	031526	034460	021451		
5203	031532	022445	047514	044507	MSLT34: .ASCII /%%LOGIC TEST 34: PES(M8911)#/
5204	031540	020103	042524	052123	
5205	031546	031440	035064	050040	
5206	031554	051505	046450	034470	
5207	031562	030461	021451		
5208	031566	022445	047514	044507	MSLT35: .ASCII /%%LOGIC TEST 35: TCW(M8903 M8905)#/
5209	031574	020103	042524	052123	
5210	031602	031440	035065	052040	
5211	031610	053503	046450	034470	
5212	031616	031460	046440	034470	
5213	031624	032460	021451		
5214	031630	022445	047514	044507	MSLT36: .ASCII /%%LOGIC TEST 36: FCS(M8903 M8905)#/
5215	031636	020103	042524	052123	
5216	031644	031440	035066	043040	
5217	031652	051503	046450	034470	
5218	031660	031460	046440	034470	
5219	031666	032460	021451		
5220	031672	022445	047514	044507	MSLT37: .ASCII /%%LOGIC TEST 37: ACCL(M8903 M8905)#/
5221	031700	020103	042524	052123	
5222	031706	031440	035067	040440	
5223	031714	041503	024114	034115	
5224	031722	030071	020063	034115	
5225	031730	030071	024465	043	
5226	031735	045	046045	043517	MSLT40: .ASCII /%%LOGIC TEST 40: PE TAPE MARK(M8902)#/
5227	031742	041511	052040	051505	
5228	031750	020124	030064	020072	
5229	031756	042520	052040	050101	
5230	031764	020105	040515	045522	
5231	031772	046450	034470	031060	
5232	032000	021451			
5233	032002	022445	047514	044507	MSLT41: .ASCII /%%LOGIC TEST 41: NRZ TAPE MARK (M8904)#/
5234	032010	020103	042524	052123	
5235	032016	032040	035061	047040	
5236	032024	055122	052040	050101	
5237	032032	020105	040515	045522	
5238	032040	024040	034115	030071	
5239	032046	024464	043		
5240	032051	045	046045	043517	MSLT42: .ASCII /%%LOGIC TEST 42: WRAP 3,NRZ,NORMAL,ODD#/
5241	032056	041511	052040	051505	
5242	032064	020124	031064	020072	
5243	032072	051127	050101	031440	
5244	032100	047054	055122	047054	
5245	032106	051117	040515	026114	
5246	032114	042117	021504		
5247	032120	022445	047514	044507	MSLT43: .ASCII /%%LOGIC TEST 43: WRAP 3,PE,NORMAL,ODD#/
5248	032126	020103	042524	052123	
5249	032134	032040	035063	053440	
5250	032142	040522	020120	026063	
5251	032150	042520	047054	051117	



5252	032156	040515	026114	042117	
5253	032164	021504			
5254	032166	022445	047514	044507	MSLT44: .ASCII /%%LOGIC TEST 44: WRAP 2,NRZ,NORMAL,ODD#/
5255	032174	020103	042524	052123	
5256	032202	032040	035064	053440	
5257	032210	040522	020120	026062	
5258	032216	051116	026132	047516	
5259	032224	046522	046101	047454	
5260	032232	042104	043		
5261	032235	045	046045	043517	MSLT45: .ASCII /%%LOGIC TEST 45: WRAP 2,PE,NORMAL,ODD#/
5262	032242	041511	052040	051505	
5263	032250	020124	032464	020072	
5264	032256	051127	050101	031040	
5265	032264	050054	026105	047516	
5266	032272	046522	046101	047454	
5267	032300	042104	043		
5268	032303	045	046045	043517	MSLT46: .ASCII /%%LOGIC TEST 46: WRAP 1,NRZ,NORMAL,ODD#/
5269	032310	041511	052040	051505	
5270	032316	020124	033064	020072	
5271	032324	051127	050101	030440	
5272	032332	047054	055122	047054	
5273	032340	051117	040515	026114	
5274	032346	042117	021504		
5275	032352	022445	047514	044507	MSLT47: .ASCII /%%LOGIC TEST 47: WRAP 1,PE,NORMAL,ODD#/
5276	032360	020103	042524	052123	
5277	032366	032040	035067	053440	
5278	032374	040522	020120	026061	
5279	032402	042520	047054	051117	
5280	032410	040515	026114	042117	
5281	032416	021504			
5282	032420	022445	047514	044507	MSLT50: .ASCII /%%LOGIC TEST 50: WRAP 0,NRZ,NORMAL,ODD#/
5283	032426	020103	042524	052123	
5284	032434	032440	035060	053440	
5285	032442	040522	020120	026060	
5286	032450	051116	026132	047516	
5287	032456	046522	046101	047454	
5288	032464	042104	043		
5289	032467	045	046045	043517	MSLT51: .ASCII /%%LOGIC TEST 51: WRAP 0,PE,NORMAL,ODD#/
5290	032474	041511	052040	051505	
5291	032502	020124	030465	020072	
5292	032510	051127	050101	030040	
5293	032516	050054	026105	047516	
5294	032524	046522	046101	047454	
5295	032532	042104	043		
5296	032535	045	046045	043517	MSLT52: .ASCII /%%LOGIC TEST 52: CORE DUMP WRITE (M8906)#/
5297	032542	041511	052040	051505	
5298	032550	020124	031065	020072	
5299	032556	047503	042522	042040	
5300	032564	046525	020120	051127	
5301	032572	052111	020105	046450	
5302	032600	034470	033060	021451	
5303	032606	022445	047514	044507	MSLT53: .ASCII /%%LOGIC TEST 53: CORE DUMP READ (M8906)#/
5304	032614	020103	042524	052123	
5305	032622	032440	035063	041440	
5306	032630	051117	020105	052504	
5307	032636	050115	051040	040505	



5308	032644	020104	046450	034470	
5309	032652	033060	021451		
5310	032656	022445	047514	044507	MSLT54: .ASCII /%%LOGIC TEST 54: EVEN PARITY WRITE (M8903 M8904)##/
5311	032664	020103	042524	052123	
5312	032672	032440	035064	042440	
5313	032700	042526	020116	040520	
5314	032706	044522	054524	053440	
5315	032714	044522	042524	024040	
5316	032722	034115	030071	020063	
5317	032730	034115	030071	024464	
5318	032736	043			
5319	032737	045	046045	043517	MSLT55: .ASCII /%%LOGIC TEST 55: EVEN PARITY READ(M8903 M8904)##/
5320	032744	041511	052040	051505	
5321	032752	020124	032465	020072	
5322	032760	053105	047105	050040	
5323	032766	051101	052111	020131	
5324	032774	042522	042101	046450	
5325	033002	034470	031460	046440	
5326	033010	034470	032060	021451	
5327	033016	022445	047514	044507	MSLT56: .ASCII /%%LOGIC TEST 56: READ REVERSE(M8906)##/
5328	033024	020103	042524	052123	
5329	033032	032440	035066	051040	
5330	033040	040505	020104	042522	
5331	033046	042526	051522	024105	
5332	033054	034115	030071	024466	
5333	033062	043			
5334	033063	045	046045	043517	MSLT57: .ASCII /%%LOGIC TEST 57: CRC(M8904)##/
5335	033070	041511	052040	051505	
5336	033076	020124	033465	020072	
5337	033104	051103	024103	034115	
5338	033112	030071	024464	043	
5339	033117	045	046045	043517	MSLT60: .ASCII /%%LOGIC TEST 60: LRC(M8904)##/
5340	033124	041511	052040	051505	
5341	033132	020124	030066	020072	
5342	033140	051114	024103	034115	
5343	033146	030071	024464	043	
5344	033153	045	046045	043517	MSLT61: .ASCII /%%LOGIC TEST 61: CORRECTABLE DATA (M8902 M8901)##/
5345	033160	041511	052040	051505	
5346	033166	020124	030466	020072	
5347	033174	047503	051122	041505	
5348	033202	040524	046102	020105	
5349	033210	040504	040524	024040	
5350	033216	034115	030071	020062	
5351	033224	034115	030071	024461	
5352	033232	043			
5353	033233	045	046045	043517	MSLT62: .ASCII /%%LOGIC TEST 62: INCORRECTABLE DATA (M8902 M8904)##/
5354	033240	041511	052040	051505	
5355	033246	020124	031066	020072	
5356	033254	047111	047503	051122	
5357	033262	041505	040524	046102	
5358	033270	020105	040504	040524	
5359	033276	024040	034115	030071	
5360	033304	020062	034115	030071	
5361	033312	024464	043		
5362	033315	045	046045	043517	MSLT63: .ASCII /%%LOGIC TEST 63: PEF(M8902)##/
5363	033322	041511	052040	051505	

5364	033330	020124	031466	020072
5365	033336	042520	024106	034115
5366	033344	030071	024462	043
5367	033351	045	046045	043517
5368	033356	041511	052040	051505
5369	033364	020124	032066	020072
5370	033372	041506	047440	042526
5371	033400	043122	047514	020127
5372	033406	046450	034470	032460
5373	033414	021451		

MSLT64: .ASCII /%LOGIC TEST 64: FC OVERFLOW (M8905)#/

```
5374
5375                                     ;MANUAL INSTRUCTION*****
5376
5377 033416 052045 050131 020105 MMSG0: .ASCII /%TYPE CR WHEN READY;#/
5378 033424 051103 053440 042510
5379 033432 020116 042522 042101
5380 033440 035531 043
5381 033443 045 046445 052517 MMSG1: .ASCII /%%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE:#/
5382 033450 052116 052040 050101
5383 033456 020105 044527 044124
5384 033464 047040 020117 051127
5385 033472 052111 020105 044522
5386 033500 043516 020054 047514
5387 033506 042101 052040 020117
5388 033514 047502 026124 051440
5389 033522 052105 052040 020117
5390 033530 047117 046040 047111
5391 033536 035105 043
5392 033541 045 042523 020124 MMSG2: .ASCII /%SET TO OFFLINE:#/
5393 033546 047524 047440 043106
5394 033554 044514 042516 021472
5395 033562 046445 053117 020105 MMSG3: .ASCII /%MOVE FORWARD TO EOT, ONLINE:#/
5396 033570 047506 053522 051101
5397 033576 020104 047524 042440
5398 033604 052117 020054 047117
5399 033612 044514 042516 021472
5400 033620 047445 043106 046040 MMSG4: .ASCII /%OFF LINE REVERSE PAST EOT, INSERT WRITE RING, ON LINE#/
5401 033626 047111 020105 042522
5402 033634 042526 051522 020105
5403 033642 040520 052123 042440
5404 033650 052117 020054 047111
5405 033656 042523 052122 053440
5406 033664 044522 042524 051040
5407 033672 047111 026107 047440
5408 033700 020116 044514 042516
5409 033706 043
5410 033707 045 046445 053117 MMSG5: .ASCII /%%MOVE TAPE TO BOT; ON LINE#/
5411 033714 020105 040524 042520
5412 033722 052040 020117 047502
5413 033730 035524 047440 020116
5414 033736 044514 042516 043
```



```

5415
5416
5417
5418 033743 045 046123 020101 TMS1: .ASCII /%SLA #/
5419 033750 043
5420 033751 045 047502 020124 TMS2: .ASCII /%BOT #/
5421 033756 043
5422 033757 045 046524 021440 TMS3: .ASCII /%TM #/
5423 033764 044445 041104 021440 TMS4: .ASCII /%IDB #/
5424 033772 051445 053504 020116 TMS5: .ASCII /%SDWN #/
5425 034000 043
5426 034001 045 042520 020123 TMS6: .ASCII /%PES #/
5427 034006 043
5428 034007 045 051523 020103 TMS7: .ASCII /%SSC #/
5429 034014 043
5430 034015 045 051104 020131 TMS8: .ASCII /%DRY #/
5431 034022 043
5432 034023 045 050104 020122 TMS9: .ASCII /%DPR #/
5433 034030 043
5434 034031 045 052116 020114 TMS10: .ASCII /%NTL #/
5435 034036 043
5436 034037 045 047505 020124 TMS11: .ASCII /%EOT #/
5437 034044 043
5438 034045 045 051127 020114 TMS12: .ASCII /%WRL #/
5439 034052 043
5440 034053 045 047515 020114 TMS13: .ASCII /%KOL #/
5441 034060 043
5442 034061 045 044520 020120 TMS14: .ASCII /%PIP #/
5443 034066 043
5444 034067 045 051105 020122 TMS15: .ASCII /%ERR #/
5445 034074 043
5446 034075 045 052101 020101 TMS16: .ASCII /%ATA #/
5447 034102 043
5448 034103 045 046111 020106 TMS17: .ASCII /%ILF #/
5449 034110 043
5450 034111 045 046111 020122 TMS18: .ASCII /%ILR #/
5451 034116 043
5452 034117 045 046522 020122 TMS19: .ASCII /%RMR #/
5453 034124 043
5454 034125 045 050103 051101 TMS20: .ASCII /%CPAR #/
5455 034132 021440
5456 034134 043045 052115 021440 TMS21: .ASCII /%FMT #/
5457 034142 042045 040520 020122 TMS22: .ASCII /%DPAR #/
5458 034150 043
5459 034151 045 047111 020103 TMS23: .ASCII /%INC #/
5460 034156 043
5461 034157 045 050126 020105 TMS24: .ASCII /%VPE #/
5462 034164 043
5463 034165 045 042520 020106 TMS25: .ASCII /%PEF #/
5464 034172 043
5465 034173 045 051114 020103 TMS26: .ASCII /%LRC #/
5466 034200 043
5467 034201 045 051516 020107 TMS27: .ASCII /%NSG #/
5468 034206 043
5469 034207 045 041506 020105 TMS28: .ASCII /%FCE #/
5470 034214 043

```

5471	034215	045	051503	021440	TMS29:	.ASCII	/%S #/
5472	034222	044445	046524	021440	TMS30:	.ASCII	/%ITM #/
5473	034230	047045	043105	021440	TMS31:	.ASCII	/%NEF #/
5474	034236	042045	042524	021440	TMS32:	.ASCII	/%DTE #/
5475	034244	047445	044520	021440	TMS33:	.ASCII	/%OPI #/
5476	034252	053445	044522	042524	TMS33A:	.ASCII	/%WRITE OPI #/
5477	034260	047440	044520	021440			
5478	034266	051045	040505	020104	TMS33B:	.ASCII	/%READ OPI #/
5479	034274	050117	020111	043			
5480	034301	045	047125	020123	TMS34:	.ASCII	/%UNS #/
5481	034306	043					
5482	034307	045	047503	051122	TMS35:	.ASCII	/%CORR #/
5483	034314	021440					
5484	034316	041445	041522	021440	TMS36:	.ASCII	/%CRC #/
5485	034324	052045	053503	021440	TMS37:	.ASCII	/%TCW #/
5486	034332	043045	051503	021440	TMS38:	.ASCII	/%FCS #/
5487	034340	040445	041503	020114	TMS39:	.ASCII	/%ACCL #/
5488	034346	043					
5489							
5490	034350					.EVEN	
5491						:WRITE BUFFER	
5492							
5493	034350	000100			WDATA:	.REPT	100
5494						-1	
5495						.ENDR	
5496							
5497							
5498						:READ BUFFER	
5499							
5500	034550	000100			RDATA:	.REPT	100
5501						0	
5502						.ENDR	
5503							
5504						:WRAP AROUND MESSAGES*****	
5505							
5506	034750	051445	052105	050125	WMSG2:	.ASCII	/%SETUP ERROR%/
5507	034756	042440	051122	051117			
5508	034764	021445					
5509	034766	050045	052101	047122	WMSG3:	.ASCII	/%PATRN NUMBER = #/
5510	034774	047040	046525	042502			
5511	035002	020122	020075	043			
5512	035007	045	047516	026516	WMSG4:	.ASCII	/%NON-EXISTANT DRIVE%/
5513	035014	054105	051511	040524			
5514	035022	052116	042040	044522			
5515	035030	042526	021445				
5516	035034	041445	030523	021440	WMSG6:	.ASCII	/%CS1 #/
5517	035042	053445	020103	043	WMSG6A:	.ASCII	/%WC #/
5518	035047	045	040502	021440	WMSG6B:	.ASCII	/%BA #/
5519	035054	043045	020103	043	WMSG6C:	.ASCII	/%FC #/
5520	035061	045	051503	020062	WMSG6D:	.ASCII	/%CS2 #/
5521	035066	043					
5522	035067	045	051504	021440	WMSG6E:	.ASCII	/%DS #/
5523	035074	042445	020122	043	WMSG6F:	.ASCII	/%ER #/
5524	035101	045	051501	021440	WMSG6G:	.ASCII	/%AS #/
5525	035106	041445	020103	043	WMSG6H:	.ASCII	/%CC #/
5526	035113	045	041104	021440	WMSG6I:	.ASCII	/%DB #/



5527	035120	046445	020122	043	WMSG6J: .ASCII	/%MR #/
5528	035125	045	052104	021440	WMSG6K: .ASCII	/%DT #/
5529	035132	052045	020103	043	WMSG6L: .ASCII	/%TC #/
5530	035137	045	047123	021440	WMSG6M: .ASCII	/%SN #/
5531	035144	041045	042101	042040	WMSG16: .ASCII	/%BAD DATA#/
5532	035152	052101	021501			
5533	035156	043445	020072	043	WMSG17: .ASCII	/%G: #/
5534	035163	045	035102	021440	WMSG20: .ASCII	/%B: #/
5535	035170	041445	035116	021440	WMSG21: .ASCII	/%CN: #/
5536	035176	041045	042101	051440	WMSG23: .ASCII	/%BAD STATUS#/
5537	035204	040524	052524	021523		
5538	035212	047045	020117	047111	WMSG24: .ASCII	/%NO INTERRUPT#/
5539	035220	042524	051122	050125		
5540	035226	021524				
5541	035230	047045	020117	046103	WMSG25: .ASCII	/%NO CLOCK UP#/
5542	035236	041517	020113	050125		
5543	035244	043				
5544	035245	045	047516	041440	WMSG26: .ASCII	/%NO CLOCK DOWN#/
5545	035252	047514	045503	042040		
5546	035260	053517	021516			
5547	035264	042045	052101	020101	WMSG27: .ASCII	/%DATA PAT: #/
5548	035272	040520	035124	043		
5549	035277	045	040502	020104	WMSG28: .ASCII	/%BAD PREAMBLE#/
5550	035304	051120	040505	041115		
5551	035312	042514	043			
5552	035315	045	040502	020104	WMSG29: .ASCII	/%BAD POSTAMBLE#/
5553	035322	047520	052123	046501		
5554	035330	046102	021505			
5555	035334	042445	051117	041440	WMSG31: .ASCII	/%EOR CLEAR DID NOT CLEAR GO%#/
5556	035342	042514	051101	042040		
5557	035350	042111	047040	052117		
5558	035356	041440	042514	051101		
5559	035364	043440	022517	043		
5560	035371	040	040520	051124	WMSG32: .ASCII	/ PATRN #/
5561	035376	020116	043			
5562						
5563		035402				
5564	035402	000000			PRE: 0	.EVEN
5565		000050				.REPT 50
5566					0	
5567						.ENDR
5568	035524	000000			POST: 0	
5569		000050				.REPT 50
5570					0	
5571						.ENDR
5572	035646	000000			WBUFF: 0	
5573		036260				.+.410
5574	036260	000000			RBUFF: 0	
5575						
5576		000001				.END



ADDFL	000746	DCHKC	020332	EXEC	016666	LTGER2	023640	LT14X	005270
AS	000526	DCHKD	020340	EXFL	000716	LTGER3	022460	LT14XX	005276
ASF	000730	DCHKE	020706	EXW2	017316	LTGX	022746	LT15	005316
ATAF	000720	DCHKX	020644	EXW2A	017356	LTGXA	022756	LT15A	005342
ATTA	023442	DCHKX1	020664	EXW2B	017374	LTGXX	023000	LT15IT	005360
ATTB	023464	DCHK0	020342	EXW2C	017402	LTG1A	023670	LT15X	005426
ATTC	023506	DCHK1	020374	EXW2E	017414	LTG1B	023706	LT15XX	005434
ATTD	023516	DCHK1A	020432	EXW2F	017422	LTG1C	024016	LT16	005440
ATTN	023420	DCHK2	020522	EXW2G	017452	LTG1T	023766	LT16A	005464
ATTP	023526	DCHK2A	020566	EXW2H	017464	LTG1X	024022	LT16IT	005502
ATTPA	023554	DCHK2B	020570	EXW2J	017476	LTG1XX	024056	LT16X	005550
ATTX	023600	DCHK3	020614	EXW2JO	017516	LTG1X1	024034	LT16XX	005556
ATTXA	023610	DCKEP	021256	EXW2J1	017520	LT1	002214	LT17	005562
ATXX	023632	DCKEPO	021310	EXW2K	017540	LT1A	002316	LT17A	005606
BA	000514	DCKE0	020724	EXW2X	017556	LT1B	002356	LT17IT	005624
BOTT	017254	DCKE1	020734	EX0	016710	LT1C	002370	LT17X	005704
BOTTA	017300	DCKE10	021216	EX1	016724	LT1ER	002400	LT17XX	005712
BOTTX	017314	DCKE11	021242	EX1A	016740	LT1ER1	002410	LT2	002444
CC	000530	DCKE12	021252	EX2	016760	LT1ER2	002416	LT2A	002506
CRCNT	000774	DCKE2	020744	EX3	017004	LT1G	002240	LT2B	002526
CS	000520	DCKE3	020754	EX5	017036	LT1G0	002230	LT2C	002544
C1	000510	DCKE4	020776	EX5A	017042	LT1X	002440	LT2ERG	002636
DATAD	000760	DCKE5	021010	EX5A1	017076	LT10	004206	LT2ER1	002560
DATA0	001032	DCKE6	021046	EX5B	017132	LT10A	004224	LT2ER2	002600
DATA1	001034	DCKE7	021104	EX5C	017166	LT10A1	004222	LT2ER3	002620
DATA2	001036	DERFL	000766	EX5CO	017232	LT10B	004264	LT2IT	002446
DATA3	001040	DOUT	025140	EX5C1	017236	LT10E1	004310	LT2LP	002654
DATBL	001030	DOUTD	025222	EX5D	017242	LT10IT	004210	LT2X	002662
DATC	000754	DOUT1	025154	FC	000516	LT10X	004336	LT20	005716
DAT1	016026	DOUT2	025202	FUN	000752	LT11	004350	LT20A	005746
DAT1A	016032	DOUT3	025210	HDRFL	000620	LT11B	004402	LT20B	006024
DAT1B	016036	DRVCLR	023152	ILFT	000544	LT11C	004450	LT20C	006036
DAT2	016050	DRVN	000624	INIT1	024206	LT11E1	004474	LT20IT	005734
DAT3	016056	DRVTP	000604	INIT2	024216	LT11E2	004530	LT20X	006052
DAT4	016066	DS	000522	INMT	024166	LT11E3	004564	LT21	006070
DAT4A	016074	DSUP	015640	INST	024234	LT11IT	004352	LT21A	006176
DB	000532	DS0	015732	ITAMT	000606	LT11X	004614	LT21B	006210
DCA	023204	DS1	015756	ITCNT	000702	LT12	004626	LT21IT	006106
DCA0	023220	DS2	015776	ITER	024122	LT12B	004656	LT21XA	006220
DCB	023260	DS3	016006	ITER0	024134	LT12C	004722	LT22	006240
DCC	023270	DS4	016016	ITER1	024142	LT12E1	004744	LT22A	006344
DCD	023160	DT	000536	ITRLP	000714	LT12E2	004770	LT22IT	006256
DCE	023300	EMADDR	000622	LTADD	000742	LT12E3	005014	LT22X	006356
DCEA	023330	EORP	022262	LTGA	022522	LT12IT	004630	LT23	006372
DCEX	023350	EORPA	022256	LTGA1	022540	LT12X	005034	LT23A	006474
DCEXA	023360	EORPX	022450	LTGA2	022572	LT13	005046	LT23IT	006410
DCEXX	023402	EORP1	022316	LTGB	022620	LT13A	005112	LT23X	006504
DCHK	020054	EORP1A	022364	LTGC	022624	LT13E1	005116	LT24	006526
DCHKA	020132	EORP2	022406	LTGCO	022712	LT13IT	005060	LT24A	006640
DCHKA0	020106	EORP3	022432	LTGD	022716	LT13X	005146	LT24B	006654
DCHKA1	020162	ER	000524	LTGER	022500	LT14	005160	LT24B0	006676
DCHKA2	020210	ERADD	000672	LTGER0	022472	LT14A	005204	LT24C	006710
DCHKB	020262	ERRF	000726	LTGER1	023646	LT14IT	005222	LT24D	006772

LT24IT	006542	LT34	010640	LT45	012370	LT6X	004042	MMSG2	033541
LT24X	007016	LT34A	010700	LT45A	012404	LT60	014146	MMSG3	033562
LT25	007052	LT34A1	010666	LT46	012460	LT60C	014242	MMSG4	033620
LT25A	007160	LT34B	010730	LT46A	012526	LT60D	014256	MMSG5	033707
LT25IT	007060	LT34C	010742	LT46B	012540	LT60E	014312	MR	000534
LT25X	007170	LT34IT	010662	LT47	012600	LT60F	014344	MSG1	025350
LT26	007204	LT34X	010772	LT47A	012614	LT60IT	014170	MSG10	026127
LT26A	007270	LT34XX	010776	LT5	003346	LT60X	014364	MSG11	026153
LT26B	007314	LT35	011002	LT5A	003370	LT61	014402	MSG12	026200
LT26C	007370	LT35A	011104	LT5B	003414	LT61A	014476	MSG13	026207
LT26IT	007212	LT35IT	011016	LT5C	003426	LT61A1	014510	MSG14	026216
LT26W	007220	LT35X	011144	LT5D	003440	LT61B	014520	MSG15	026233
LT26X	007404	LT36	011154	LT5E	003464	LT61C	014534	MSG15A	026251
LT27	007420	LT36A	011224	LT5ER1	003500	LT61D	014564	MSG16	026277
LT27A	007454	LT36IT	011170	LT5ER2	003526	LT61E	014566	MSG17	026307
LT27B	007512	LT36X	011264	LT5IT	003356	LT61F	014630	MSG18	026323
LT27IT	007444	LT37	011274	LT5X	003554	LT61IT	014432	MSG19	026342
LT27X	007522	LT37A	011350	LT5XX	003614	LT61X	014650	MSG2	025454
LT27XX	007532	LT37B	011402	LT50	012674	LT61XX	014662	MSG2A	025473
LT3	002674	LT37IT	011310	LT50A	012742	LT62	014666	MSG20	026361
LT3A	002714	LT37X	011434	LT50B	012754	LT62A	015002	MSG21	026413
LT3B	002734	LT4	003060	LT51	013012	LT62B	015016	MSG22	026451
LT3C	002750	LT4A	003154	LT51A	013026	LT62D	015046	MSG23	026472
LT3ER1	002762	LT4B	003222	LT52	013102	LT62E	014760	MSG24	026525
LT3ER2	003010	LT4C	003236	LT52A	013162	LT62E1	014772	MSG25	026553
LT3IT	002676	LT4D	003256	LT52X	013212	LT62F	015100	MSG25A	026602
LT3X	003032	LT4ERG	003312	LT53	013224	LT62IT	014714	MSG26	026627
LT3XX	003050	LT4ER1	003272	LT53A	013312	LT62X	015124	MSG27	026656
LT30	007536	LT4ER2	003304	LT53X	013350	LT62XX	015134	MSG3	025622
LT30A	007640	LT4G	003104	LT54	013360	LT63	015140	MSG30	026672
LT30B	007702	LT4G0	003074	LT55	013436	LT63A	015232	MSG31	026700
LT30C	007742	LT4X	003342	LT56	013514	LT63B	015276	MSG32	026716
LT30D	007760	LT40	011444	LT57	013600	LT63C	015320	MSG33	026734
LT30E	010010	LT40A	011526	LT57A	013704	LT63D	015354	MSG34	026755
LT30IT	007560	LT40IT	011466	LT57B	013720	LT63IT	015170	MSG35	026772
LT30X	010034	LT40W	011474	LT57B1	013750	LT63X	015400	MSG36	027010
LT31	010054	LT40X	011562	LT57B2	013770	LT63XX	015412	MSG37	027026
LT31A	010150	LT40XX	011566	LT57C	014042	LT630	015416	MSG4	025644
LT31B	010212	LT41	011572	LT57D	014056	LT64	015420	MSG40	027044
LT31C	010230	LT41A	011650	LT57E	014110	LT64A	015512	MSG41	027050
LT31D	010276	LT41B	011704	LT57IT	013632	LT64B	015524	MSG42	027067
LT31E	010346	LT41C	011734	LT57PS	013612	LT64C	015530	MSG43	027104
LT31IT	010070	LT41D	011766	LT57X	014130	LT64IT	015436	MSG44	027210
LT31W	010104	LT41IT	011606	LT6	003626	LT64X	015602	MSG45	027232
LT31W1	010236	LT41X	012006	LT6A	003644	LT7	004054	MSG46	027254
LT31X	010362	LT42	012040	LT6A1	003642	LT7A	004072	MSG47	027360
LT32	010404	LT42A	012106	LT6B	003650	LT7B	004116	MSG5	025662
LT32A	010512	LT42B	012120	LT6C	003676	LT7C	004062	MSG50	027412
LT32IT	010420	LT43	012162	LT6D	003714	LT7ER1	004142	MSG51	027431
LT32X	010522	LT43A	012176	LT6ER1	003740	LT7IT	004056	MSG52	027460
LT33	010536	LT44	012252	LT6ER2	003766	LT7X	004170	MSG53	027474
LT33IT	010552	LT44A	012320	LT6ER4	004032	MMSG0	033416	MSG54	027506
LT33X	010630	LT44B	012332	LT6IT	003630	MMSG1	033443	MSG55	027517



MSG56	027567	MSLT61	033153	SERFL	000772	TMS15	034067	TR15	000660
MSG6	025702	MSLT62	033233	SERNUM	000602	TMS16	034075	TSCD	001732
MSG7	025721	MSLT63	033315	SETPES	005302	TMS17	034103	TSCD0	001764
MSG8	025737	MSLT64	033351	SETUP	016340	TMS18	034111	TSCD1	002020
MSG8A	025756	MSLT7	030220	SET0	016354	TMS19	034117	TSCD2	002032
MSG9	026105	MTINT	024302	SET1	016360	TMS2	033751	TSCD3	002050
MSLT1	027614	NRZOF	000662	SET2	016414	TMS20	034125	TSTTBL	001066
MSLT10	030264	OCTP	024712	SKAT	001014	TMS21	034134	TTIN	024512
MSLT11	030337	OCTPE	024702	SLAF	000722	TMS22	034142	TTINT	024316
MSLT12	030404	OCTPE1	024716	SLVN	000664	TMS23	034151	TTIN1	024532
MSLT13	030455	OCTPG	025074	SN	000540	TMS24	034157	TTIN2	024546
MSLT14	030522	OCTPG0	025112	SNPG	025324	TMS25	034165	TTOUT	024564
MSLT15	030570	OCTPG1	025116	SNPT	025246	TMS26	034173	TTR	024326
MSLT16	030636	OCTP0	024740	SPO	016472	TMS27	034201	TTR0	024334
MSLT17	030704	OCTP1	024762	SP01	016460	TMS28	034207	TTR1	024362
MSLT2	027670	OCTP2	024770	SP1	016476	TMS29	034215	TTR1A	024376
MSLT20	030752	OCTP3	025060	SP1A	016542	TMS3	033757	TTR1B	024412
MSLT21	031031	OFL	025136	SP1B	016562	TMS30	034222	TTR2	024450
MSLT22	031065	PATRN	001002	SP3	016632	TMS31	034230	TTR3	024460
MSLT23	031122	PCNTR	001016	SP4	016652	TMS32	034236	TTR4	024470
MSLT24	031164	PD0	021370	SP5	016664	TMS33	034244	TTR5	024472
MSLT25	031224	PD1	021404	SSCF	000724	TMS33A	034252	T1AD	001072
MSLT26	031260	PD2	021446	START	001414	TMS33B	034266	T1IAD	001074
MSLT27	031314	PD3	021452	STATC	001012	TMS34	034301	T10AD	001126
MSLT3	027747	PD4	021462	STATF	001004	TMS35	034307	T10IAD	001130
MSLT30	031350	PD5	021602	STATIC	015614	TMS36	034316	T11AD	001132
MSLT31	031406	PD6	021626	STATX	015636	TMS37	034324	T11IAD	001134
MSLT32	031442	PEXFL	000736	STFLG	000740	TMS38	034332	T12AD	001136
MSLT33	031476	PFLG	000666	STSCD	002072	TMS39	034340	T12IAD	001140
MSLT34	031532	POST	035524	STO	001606	TMS4	033764	T13AD	001142
MSLT35	031566	PPGEN	022204	ST1	001630	TMS5	033772	T13IAD	001144
MSLT36	031630	PPGO	022220	ST2	001726	TMS6	034001	T14AD	001146
MSLT37	031672	PPG1	022244	SWR	000570	TMS7	034007	T14IAD	001150
MSLT4	030031	PRCHK	021352	TADX	001412	TMS8	034015	T15AD	001152
MSLT40	031735	PRE	035402	TC	000542	TMS9	034023	T15IAD	001154
MSLT41	032002	PREFL	000770	TCRLF	024616	TOB	000614	T16AD	001156
MSLT42	032051	PSCHK	021316	TCRLFA	024634	TOG	024664	T16IAD	001160
MSLT43	032120	PSW	000566	TEMP1	000674	TPB	000600	T17AD	001162
MSLT44	032166	RBUFF	036260	TEMP2	000676	TPS	000576	T17IAD	001164
MSLT45	032235	RCDP	001010	TEMP3	000700	TREF	000734	T2AD	001076
MSLT46	032303	RDAD	000762	TEND	002144	TRO0	000626	T2IAD	001100
MSLT47	032352	RDATA	034550	TENDX	002176	TRO1	000630	T20AD	001166
MSLT5	030110	RDRVF	001006	TEX	024700	TRO2	000632	T20IAD	001170
MSLT50	032420	REGP	023016	TIB	000616	TRO3	000634	T21AD	001172
MSLT51	032467	REGS	000612	TINER	024474	TRO4	000636	T21IAD	001174
MSLT52	032535	RTRN	000670	TKB	000574	TRO5	000640	T22AD	001176
MSLT53	032606	SAV1	000704	TKS	000572	TRO6	000642	T22IAD	001200
MSLT54	032656	SAV2	000706	TMS1	033743	TRO7	000644	T23AD	001202
MSLT55	032737	SAV3	000710	TMS10	034031	TR10	000646	T23IAD	001204
MSLT56	033016	SCF	000732	TMS11	034037	TR11	000650	T24AD	001206
MSLT57	033063	SCOLP	000712	TMS12	034045	TR12	000652	T24FL	000744
MSLT6	030154	SCOPE	024072	TMS13	034053	TR13	000654	T24IAD	001210
MSLT60	033117	SCOPE1	024106	TMS14	034061	TR14	000656	T25AD	001212



T25IAD	001214	T40IAD	001270	T63AD	001402	WMSG16	035144	WMSG6J	035120
T26AD	001216	T41AD	001272	T63IAD	001404	WMSG17	035156	WMSG6K	035125
T26IAD	001220	T41IAD	001274	T64AD	001406	WMSG2	034750	WMSG6L	035132
T27AD	001222	T42AD	001276	T64IAD	001410	WMSG20	035163	WMSG6M	035137
T27IAD	001224	T43AD	001302	T7AD	001122	WMSG21	035170	WPGFL	001000
T3AD	001102	T44AD	001306	T7IAD	001124	WMSG23	035176	WSTCK	017562
T3IAD	001104	T45AD	001312	UDES	000776	WMSG24	035212	WSTG	017730
T30AD	001226	T46AD	001316	VECT	000610	WMSG25	035230	WSTGO	020000
T30IAD	001230	T47AD	001322	WAM	000750	WMSG26	035245	WSTX	020052
T31AD	001232	T5AD	001112	WAMO	016110	WMSG27	035264	WTAD	000756
T31IAD	001234	T5IAD	001114	WAM01	016116	WMSG28	035277	W1DCHK	021652
T32AD	001236	T50AD	001326	WAM1	016154	WMSG29	035315	W1D0	021702
T32IAD	001240	T51AD	001332	WAM2	016166	WMSG3	034766	W1D0A	021720
T33AD	001242	T52AD	001336	WAM2A	016222	WMSG31	035334	W1D1	021726
T33IAD	001244	T53AD	001342	WAM3	016232	WMSG32	035371	W1D2	022004
T34AD	001246	T54AD	001346	WAM3A	016312	WMSG4	035007	W1D2A	022056
T34IAD	001250	T55AD	001352	WAM3B	016330	WMSG6	035034	W1D3	022146
T35AD	001252	T56AD	001356	WBUFF	035646	WMSG6A	035042	W1D4	022160
T35IAD	001254	T57AD	001362	WC	000512	WMSG6B	035047	W1D4A	022170
T36AD	001256	T57IAD	001364	WCDP0	001054	WMSG6C	035054	W1D5	022174
T36IAD	001260	T6AD	001116	WCDP2	001042	WMSG6D	035061	W2FLG	000764
T37AD	001262	T6IAD	001120	WCS1	001020	WMSG6E	035067	.	= 036262
T37IAD	001264	T6IAD	001372	WCS2	001022	WMSG6F	035074		
T4AD	001106	T61IAD	001374	WDATA	034350	WMSG6G	035101		
T4IAD	001110	T62AD	001376	WDS	001024	WMSG6H	035106		
T40AD	001266	T62IAD	001400	WER	001026	WMSG6I	035113		

. ABS. 036262 000

ERRORS DETECTED: 0

,CZTUKA.SEQ/SOL\_CZTUKA.P11  
RUN-TIME: 45 92 9 SECONDS  
RUN-TIME RATIO: 1164/147=7.9  
CORE USED: 5K (10 PAGES)