

.REM %

IDENTIFICATION

PRODUCT CODE: AC-E487B-MC
PRODUCT NAME: CZTUOB0 TM03/TU45 LGC PRT 1
DATE CREATED: 25 MAY 1978
MAINTAINER: CSS - NASHUA
AUTHOR: J. G. ADAMS/R. J. COLLINS

UPDATE INFORMATION: DATE AUTHOR
 19 OCT. 1979 JAMES DUPRE
 10 MARCH 1980 VIJAY ANANDWALA

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 RESPONSIBLITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (©) 1975, 1980 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

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

1. ABSTRACT

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

2. REQUIREMENTS (HARDWARE)

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

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE.

4. STARTING PROCEDURE

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

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

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

4.1 AUTOMATIC MODE OPERATION

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

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

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

4.2 SAMPLE START AT 200

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

PARAMETER REQUEST: <DEFAULT> (RESPONSE) [LOCATION:]

TM03-TU45 CONTROL LOGIC TEST- PART I (CZTU0B0)
ASSUME TAPE IS AT BOT
TYPE ^C TO RESTART

REGISTER START: <172440> (CR)	[REGS:]
VECTOR ADDRESS: <224> (CR)	[VECT:]
TM03 DRIVE: <0> (CR)	[DRVN:]
TU45 SLAVE: <0> (CR)	[SLVN:]
STATIC TESTS ONLY: <0> (CR)	[STATC:]
IF THE SOFTWARE SWR IS INVOKED:	
SWR = <000000> NEW - (CR)	[SWREG:]

5. CONSOLE SWITCH SETTINGS

CONTROL:

- 1) CONTROL G <^G>:
INVOKES THE SOFTWARE SWR AND ALLOWS USER TO ENTER SWITCH SETTING
THE MACHINE WILL THEN TYPE: SWR=XXXXXX NEW=
WHERE: XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.
AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
OF THE FOLLOWING AT THE TTY:
A) TYPE THE NEW SWITCH SETTING
B) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
REGISTER CONTENTS WILL NOT BE CHANGED.
- 2) CONTROL A <^A>:
ALTERNATES SWITCH REGISTER FROM HARDWARE TO SOFTWARE & VICE VERSA
- 3) CONTROL C <^C>:
RESTARTS THE PROGRAM AT 200
- 4) CONTROL U <^U>:
DELETES ALL CHARACTERS TYPED IN RESPONSE TO A REQUEST

ALL SWITCHES ARE USED (0-15) AND THE NORMAL, OR DEFAULT, RUN
IS DONE WITH ALL SWITCHES SET TO ZERO (0).
ALL SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME.

- SW15: 1=HALT ON ERROR
0=CONTINUE
- SW14: 1=LOOP ON ERROR (SCOPE)
0=CONTINUE
- SW13: 1=DO NOT PRINT ERRORS
0=PRINT ALL ERRORS
- SW12: 1=DO CONTINUOUS CYCLE
0=HALT AT END OF PASS
- SW11: 1=INHIBIT ITERATIONS
0=ITERATE EACH TEST ITS ASSIGNED AMOUNT
- SW10: 1=HALT AT END OF CURRENT TEST
0=CONTINUE TO NEXT TEST
- SW9: 1=DO MANUAL INTERVENTION TESTS
0=INHIBIT MANUAL INTERVENTION
- SW5-0: SELECT INDIVIDUAL TEST ** 00=DO ALL TESTS

6. ERROR PRINTOUTS

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

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

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

EXAMPLES:

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

LOGIC TEST 1: DRIVE ADDRESSING (M8939 OR RH)
NON-EXIST DRIVE 3 EXPT-NOT RECVD
ITER: 3

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

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

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

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

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

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

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

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

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

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

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

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

LOGIC TEST 30: DTE (M8906 RH)
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 (M8939)
NOT RESET BY DRIVE CLEAR

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

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

7. OPERATION

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

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

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

SINGLE TEST SELECTION: (SW0-SW5)

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

8. SUB TEST SUMMARIES

LOGIC TEST #1: DRIVE ADDRESSING

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

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

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8939

CIRCUITS

PRINT REFERENCES

RH-DS BITS	(CSR8)
RH-NED BIT	(CSR8)
MASSBUS CABLE C (DEM,TRA,DS BITS)	(MB3)
DRIVE ADDRESS	(MB12)
DEM-TRA HANDSHAKE	

LOGIC TEST #2: REGISTER ADDRESSING

PURPOSE: CHECK THE REGISTER SELECT LINES

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

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8939,M8905-YB,M8933

CIRCUITS

PRINT REFERENCE

C-LINES	(MB1,2,3),(MB13),(MB14),(MB15)
RH REGISTER SELECT	(BCTA)
TM03 REGISTER SELECT	(MB12)
MASSBUS REGISTER SELECT LINES	(MB1,2)
PARITY TREE	(MB14)
CPAR,ILR BITS	(MB11)

LOGIC TEST #3: CONTROL BUS

PURPOSE: VERIFY THAT ALL CONTROL LINES PROPERLY TRANSMIT

ONES AND ZEROS.

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

LIKELY FAULT LOCATIONS: M5904,CABLES,M5903YA,M8939,M8905-YB,M8933

CIRCUITS

PRINT REFERENCE

C-LINES
C-BUS MULTIPLEXERS
ERROR BIT
MCPE BIT

(MB1,2,3)
(MB1,3,4,5,8)(TCCM7)(MR)
(MB11)
(PACA)

LOGIC TEST #4: SLAVE ADDRESSING

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

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

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

LIKELY FAULTS LOCATIONS: M8905-YB,M8937,CABLE,M9001,M8928,M9001YA,M8933

CIRCUITS

PRINT REFERENCE

REGISTER SELECT
SLAVE ADDRESS BITS
SLAVE ADDRESS LINES
TU45 ADDRESS DECODE
SPR BIT

(MB12)
(MR6)
(M8937,2-2),(LAW6)
(LAW6)
(LAW6)(M9001YA)(TCCM7)

LOGIC TEST #5: MAINTENANCE REGISTER BITS

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

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

LIKELY FAULT LOCATIONS: M8905-YB

CIRCUITS

PRINT REFERENCE

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

LOGIC TEST #6: TAPE CONTROL REGISTER BITS

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

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

LIKELY FAULT LOCATIONS: M8939,M8905-YB

CIRCUITS

PRINT REFERENCE

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

LOGIC TEST #7: FRAME COUNT BIT TEST

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

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

LIKELY FAULT LOCATIONS: M8939

CIRCUITS

PRINT REFERENCE

TMO3 REGISTER SELECT	(MB12)
FRAME COUNT REGISTER	(MB18)
FRAME COUNT MULTIPLEXERS	(MB110)

LOGIC TEST #10: FUNCTION CODE BIT TEST

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

PROGRAMMED SEQUENCE: THE C1 REGISTER IS WRITTEN WITH ALL ZEROS. DATA IS CHECKED ON THE 5 FUNCTION CODE BITS (BITS 1-5). BITS 1-5 ARE WRITTEN WITH ONES, CHECK AND REPEAT WITH ALL ZEROS.

LIKELY FAULT LOCATION: M8939, M8905-YB

CIRCUITS

PRINT REFERENCE

TMO3 REGISTER SELECTION	(MB12)
FUNCTION CODE FLOPS	(MB15)
FUNCTION CODE MULTIPLEXERS	(MR6)

LOGIC TEST #11: GO BIT SET, RESET

PURPOSE: TO VERIFY THAT THE GO BIT CAN BE SET IN A SIMULATED READ OPERATION AND CLEARED WITH AN INIT.

PROGRAMMED SEQUENCE: INIT AND CHECK THAT GO=0. SET UP A SIMULATED READ OPERATION BY LOADING A WAM3 15(OCTAL) INTO THE MAINTENANCE REGISTER, CLEARING THE FRAME COUNT REGISTER TO SET FCS, LOAD 1700 (FORMAT) INTO THE TAPE CONTROL REGISTER, SETTING READ COMMAND AND GO BIT. CHECK FOR GO=1. INIT AND CHECK THAT GO BIT=0.

LIKELY FAULT LOCATION: MASSBUS CABLE B(INIT),M8939,M8905-YB

<u>CIRCUIT</u>	<u>PRINT REFERENCE</u>
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

LOGIC TEST #13: INTERRUPT TEST

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

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

LIKELY FAULT LOCATION:

CIRCUITS

PRINT REFERENCE

INTERRUPT CONTROL

BCTF

MANUAL INTERVENTION TESTS 14,15,16,17

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

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

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

LIKELY FAULT LOCATION: M8928,SLAVE CABLE, M8933

CIRCUIT

PRINT REFERENCE

MOL
WRL
DPR
DRY
BOT

LAW6,TCCM7,M8908,M9001YA,YC
LAW8,TCCM7,M8908,M9001YA,YC
TCCM7
TCCM7
LAW6,TCCM7,M8908YA,M8913,YA

LOGIC TEST #15: STATUS AT BOT,OFFLINE,LOADED, NO WRITE RING

PURPOSE: TO TEST ATA,DPR,DRY,SSC

PROGRAMMED SEQUENCE: OPERATOR IS INSTRUCTED TO TAKE DRIVE
OFFLINE: ATA,SSC,DPR,DRY ARE CHECKED.

LIKELY FAULT LOCATION: M8928,M8933,M8939,SLAVE CABLE

CIRCUIT

PRINT REFERENCE

SSC
ATA

LAW8,M8913,M8913YA,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: M8928,SLAVE CABLE,M8933

CIRCUIT

PRINT REFERENCE

SSC
EOT
SLA

LAW8,M8913,M8913YA,TCCM7
LAW6,TCCM7,M8908YA,M8913YA
LAW8,TCCM7,M9001YA,YC,M8908

LOGIC TEST #17: STATUS AT ONLINE LOADED

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

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

LOGIC TEST #20: ILLEGAL FUNCTION

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

LIKELY FAULT LOCATION: M8939

CIRCUIT

PRINT REFERENCE

SET ILF DECODE
ILF FLOP
ILF MULTIPLEXER

MB15,MB17
MB111
MB110

LOGIC TEST #21: REGISTER MODIFICATION REFUSED

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

LIKELY FAULT LOCATION: M8939

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

LOGIC TEST #22: CONTROL BUS PARITY (CPAR)

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

LIKELY FAULT LOCATIONS: M8939

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

LOGIC TEST #23: FORMAT ERROR (FMT)

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

LIKELY FAULT LOCATIONS: M8905-YB, M8906, M8939

CIRCUIT PRINT REFERENCE

FORMAT BITS	MR6
ILF DECODE	BF3
ILF FLOP	MBI11
ILF MULTIPLEXERS	MBI10

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

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

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

CIRCUIT PRINT REFERENCE

MM CLK	MRS
WRT CLK GENERATION	TCCM4
DPAR FLOP	MBI11
DATA BUS PARITY TREE	BF3

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

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

LIKELY FAULT LOCATION: M8939

<u>CIRCUIT</u>	<u>PRINT REFERENCE</u>
NEF FLOP	MB111
NEF MULTIPLEXER	MB110
SET NEF	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: M8939, MB CABLE, M8933, M8905-YB

<u>CIRCUITS</u>	<u>PRINT REFERENCE</u>
RUN LINE	MB1
EBL PLS	MB19
FCE FLOP	MB111
SHUTDOWN LOGIC	TCCM5
MAINT. FUNCTION DECODE	MR5

LOGIC TEST #27: ILLEGAL REGISTER

THIS TEST IS BYPASSED DUE TO THE REQUIREMENT OF INSERTING AND REMOVING ADDRESS JUMPERS IN THE RH FOR PROPER OPERATION.

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

LIKELY FAULT LOCATIONS: MASSBUSS, M8939

CIRCUITS

PRINT REFERENCE

REGISTER SELECT LINES	MB1, MB2
REGISTER SELECT DECODE	MBI2
ILR FLOP	MBI11

LOGIC TEST #30: DRIVE TIMING ERROR

PROGRAMMED SEQUENCE:

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

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

CIRCUITS

PRINT REFERENCES

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

LOGIC TEST 31: OPERATION INCOMPLETE (OPI)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8933, M8939

CIRCUITS

PRINT REFERENCES

OPI TIMER
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: M8939, M8928, SLAVE CABLE

CIRCUITS

PRINT REFERENCES

UNSAFE FLOP
SET UNSAFE
MOL GENERATION

MBI11
MBI7
LAW6

LOGIC TEST 33: POSITIONING IN PROGRESS (PIP)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8939, M8933

CIRCUITS

PRINT REFERENCES

SPACE FUNCTION DECODE	MB15
PIP GENERATION	TCCM7
STATUS REGISTER	TCCM7

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

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

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

CIRCUITS

PRINT REFERENCES

DENSITY BITS	MR6
DENSITY LINES	SBC
PES CIRCUIT	SC3
PES STATUS BIT	TCCM7

LOGIC TEST 35: TAPE CONTROL WRITE (TCW)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATION: M8905-YB

<u>CIRCUIT</u>	<u>PRINT REFERENCES</u>
TCW	MR6

LOGIC TEST 36: FRAME COUNTER STATUS (FCS)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8939, M8933

<u>CIRCUITS</u>	<u>PRINT REFERENCES</u>
FCS BIT FCS MULTIPLEXER	MB18 TCCM7

LOGIC TEST 37: ACCELERATION (ACCL)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8933, M8928

CIRCUITS

PRINT REFERENCES

ACCL BIT, MOTION DELAY COUNTER 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 M8932.

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

CIRCUITS

PRINT REFERENCES

TAPE MARK DETECTOR	TCPE4, TCPE5
TAPE MARK MULTIPLEXER	TCCM7
ENVELOPE SIGNALS	DS 3, 5, 7
WRITE DATA BUFFER	TCCM2
RDA MULTIPLEXERS	TCCM6
WRITE TAPE MARK FUNCTION	MB15
WAMO SIGNAL	MRS

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

PROGRAMMED SEQUENCE:

SAME AS TEST 40 EXCEPT NRZ DENSITY IS SELECTED.

LIKELY FAULT LOCATIONS: M8933, M8934

CIRCUITS

PRINT REFERENCES

WRITE DATA BUFFER
RSDO MULTIPLEXER
RDA MULTIPLEXERS
TM DETECTOR
ILLEGAL TAPE MARK FLOP

TCCM2
TCCM6
TCCM6
CNRZ4
CNRZ4

THE NEXT 5 TESTS CONSISTS OF WRITING ON TAPE USING MAINTENANCE MODE FUNCTIONS TO FORCE ERROR CONDITIONS TO CHECK THE ERROR CHECKING CAPABILITIES. OCCASIONAL ERRORS MAY RESULT FROM TAPE DEFECTS. CONSTANT ERROR MAY BE THE RESULT OF PROBLEMS WITH ERROR CHECKING CIRCUITRY OR PROBLEMS WITH THE DRIVE. DEBUG OF THE PROBLEMS MAY BE EASIER USING DATA RELIABILITY OF UTILITY DRIVER.

LOGIC TEST 42: CYCLIC REDUNDANCY ERROR

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905-YB, M8934, G056, SLAVE CABLE,
----- M8928

CIRCUITS

PRINT REFERENCES

MM FUNCTION DECODE
CRC CHECK CIRCUIT

MRS
CNRZ3

LOGIC TEST 43: LRC

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8505, M8933, M8928, M8934

CIRCUITS

PRINT REFERENCES

MM FUNCTION DECODE
WRITE LINE DRIVERS
WRITE HEAD DRIVERS
LRC CHECKING

MRS
TCCM2
LAW3, 4
CNRZ3

LOGIC TEST 44: PE CORRECTABLE DATA

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905-YB, M8901, M8932

CIRCUITS

PRINT REFERENCES

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

LOGIC TEST 45: PE INCORRECTABLE DATA

REPEAT OF TEST 44, EXCEPT THAT THE MAINT. MODE FUNCTION 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: M8932, M8901

CIRCUIT

PRINT REFERENCE

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

LOGIC TEST 46: PE FORMAT

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

LIKELY FAULT LOCATIONS: M8932, M8933, M8905-YB

CIRCUITS

PRINT REFERENCES

PEF.
WRITE BUFFER
MM DECODE

TCPE2
TCCM2
MR5

LOGIC TEST 47: FRAME COUNT OVERFLOW

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

LIKELY FAULT LOCATION: M8939

FRAME COUNT REGISTER MB18

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

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

PROGRAM SEQUENCE:

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

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

THIS TEST IS THE COMPLEMENT OF LOGIC TEST 50 ABOVE.

2499
2500
2501

.LIST BIN,LOC,SEQ
.TITLE TM03/TU45 CONTROL LOGIC TEST PART I
;CZTUOB0

2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528

```
:25 MAY 78  
:J.G. ADAMS/R. J. COLLINS  
:REVISED JUN 1977 BY J. G. ADAMS :CHANGED MODULE REFERENCES TO  
:TM03 MODULES  
.MCALL .SACT11, .SEOP, $CATCH, $SAVE, $RESTORE, $CHAIN, $CHNMODE  
.NLIST MC  
.LIST ME  
.ENABLE ABS,AMA  
  
:CONSOLE SWITCHES*****  
:  
:SW15: 1=HALT ON ERROR  
: 0=CONTINUE  
:  
:SW14: 1=LOOP ON ERROR  
: 0=CONTINUE  
:  
:SW13: 1=DO NOT PRINT ERRORS  
: 0=PRINT ERRORS  
:  
:SW12: 1=HALT AT END OF PASS  
: 0=CONTINUOUS CYCLE  
:  
:SW11: 1=INHIBIT ITERATIONS  
: 0=DO ITERATIONS  
:  
:SW10: 1=HALT AT END OF EACH TEST  
: 0=CONTINUE  
:  
:SW9: 1=DO MANUAL INTERVENTION TESTS  
: 0=INHIBIT MANUAL INTERVENTION  
:  
:SW0-5: SELECT TEST NUMBER :: 00=ALL TESTS
```



```
2577 ;REGISTER EQUIVS*****
2578
2579 R0=%0
2580 R1=%1
2581 R2=%2
2582 R3=%3
2583 R4=%4
2584 R5=%5
2585 SP=%6
2586 PC=%7
2587
2589
(1) ;ACT11 HOOK *****
(1) $SVPC=. ;SAVE CURRENT LOCATION CTR
(1) 000046 000046 .=46
(1) 000046 002566 .WORD $ENDAD ;SET LOCATION 46
(1) 000052 000052 .=52
(1) 000052 000000 .WORD 0 ;SET LOCATION 52 = 0
(1) 000764 000764 .=$SVPC ;RESTORE LOCATION CTR
2590 ;TTY INTERRUPT VECTOR*****
2591
2592 .=60
2593 000060 016760 .WORD TTINT ;TTY INTERRUPT HEADER ADDRESS
2594 000062 000340 .WORD 340 ;PRIORITY LEVEL 7
2595
2596 ;SOFTWARE SWITCH REGISTER*****
2597 ;USED IF HARDWARE SWR = 17777 OR NOT AVAILABLE
2598 000176 000176 .=176
2599 000176 000000 SWREG: .WORD 0 ;SOFTWARE SWITCH REGISTER
2600
2601 ;START ADDRESS*****
2602 .=200
2603 000200 000137 001330 JMP START ;PROGRAM START
2604
2605 ;RESTART ADDRESS*****
2606 .=210
2607 000210 000137 002066 JMP ST2
2608
2609 ;TM03 INTERRUPT VECTOR*****
2610
2611 .=224
2612 000224 016750 MTINT ;TAPE INTERRUPT HANDLER ADDRESS
2613 000226 000340 340
2614
```

```
2616
2617      000510      . = 510
2618      ; MASS BUS REGISTER EQUIVS*****
2619
2620      000510      172440      C1:      172440
2621      000512      172442      WC:      172442
2622      000514      172444      BA:      172444
2623      000516      172446      FC:      172446
2624      000520      172450      CS:      172450
2625      000522      172452      DS:      172452
2626      000524      172454      ER:      172454
2627      000526      172456      AS:      172456
2628      000530      172460      CC:      172460
2629      000532      172462      DB:      172462
2630      000534      172464      MR:      172464
2631      000536      172466      DT:      172466
2632      000540      172470      SN:      172470
2633      000542      172472      TC:      172472
2634
2635      ; ILLEGAL FUNCTION CODES
2636
2637      000544      005405      ILFT:    5405
2638      000546      007415      7415
2639      000550      016423      16423
2640      000552      020437      20437
2641      000554      022443      22443
2642      000556      025447      25447
2643      000560      031455      31455
2644      000562      033465      33465
2645      000564      036473      36473
2646
2647      ; CONSTANTS*****
2648
2649      000566      177776      PSW:     177776      ; PROCESSOR STATUS
2650      000570      177570      SWR:     177570      ; SWITCH REGISTER
2651      000572      177560      TKS:     177560      ; TTY READER STATUS
2652      000574      177562      TKB:     177562      ; TTY READ BUFFER
2653      000576      177564      TPS:     177564      ; TTY PUNCH STATUS
2654      000600      177566      TPB:     177566      ; TTY PUNCH BUFFER
2655      000602      177777      SERNUM:  177777      ; SERIAL NUMBER
2656      000604      000011      DRVTP:   011        ; DRIVE TYPE
2657      000606      000020      ITAMT:   20         ; ITERATION AMOUNT
2658      000610      000224      VECT:    224        ; INTERRUPT VECTOR(RH)
2659      000612      172440      REGS:    172440     ; STARTING REGISTER ADDRESS
```


2661
2662
2663 000614 000000
2664 000616 000000
2665 000620 000000
2666 000622 000000
2667 000624 000000
2668 000626 000000
2669 000630 000000
2670 000632 000000
2671 000634 000000
2672 000636 000000
2673 000640 000000
2674 000642 000000
2675 000644 000000
2676 000646 000000
2677 000650 000000
2678 000652 000000
2679 000654 000000
2680 000656 000000
2681 000660 000000
2682 000662 000000
2683 000664 000000
2684 000666 000000
2685 000670 000000
2686 000672 000000
2687 000674 000000
2688 000676 000000
2689 000700 000000
2690 000702 000000
2691 000704 000000
2692 000706 000000
2693 000710 000000
2694 000712 000000
2695 000714 000000
2696 000716 000000
2697 000720 000000
2698 000722 000000
2699 000724 000000
2700 000726 000000
2701 000730 000000
2702 000732 000000
2703 000734 000000
2704 000736 000000
2705 000740 000000
2706 000742 000000
2707 000744 000000
2708 000746 000000
2709 000750 000000
2710 000752 000000
2711 000754 000000
2712 000756 000000
2713 000760 000000
2714 000762 000000
2715 000764 000000
2716 000766 000000

:FLAGS AND COUNTERS*****

TOB: 0
TIB: 0
HDRFL: 0
EMADDR: 0
DRVN: 0
TROO: 0
TRO1: 0
TRO2: 0
TRO3: 0
TRO4: 0
TRO5: 0
TRO6: 0
TRO7: 0
TR10: 0
TR11: 0
TR12: 0
TR13: 0
TR14: 0
TR15: 0
NRZOF: 0
SLVN: 0
PFLG: 0
RTRN: 0
ERADD: 0
TEMP1: 0
TEMP2: 0
TEMP3: 0
ITCNT: 0
SAV1: 0
SAV2: 0
SAV3: 0
SCOLP: 0
ITRLP: 0
EXFL: 0
ATAF: 0
SLAF: 0
SSCF: 0
ERRF: 0
ASF: 0
SCF: 0
TREF: 0
PEXFL: 0
STFLG: 0
LTADD: 0
T24FL: 0
ADDFL: 0
WAM: 0
FUN: 0
DATC: 0
WTAD: 0
DATAD: 0
RDAD: 0
W2FLG: 0
DERFL: 0

2717	000770	000000	PREFL:	0
2718	000772	000000	SERFL:	0
2719	000774	000000	CRCNT:	0
2720	000776	000000	UDES:	0
2721	001000	000000	WPGFL:	0
2722	001002	000000	PATRN:	0
2723	001004	000000	STATF:	0
2724	001006	000000	RDRVF:	0
2725	001010	000000	RCDP:	0
2726	001012	000000	STATC:	0
2727	001014	000000	SKAT:	0
2728	001016	000000	PCNTR:	0
2729				;PASS COUNTER
2730				
2731				;EXPT WRAP STATUS*****
2732	001020	000000	WCS1:	0
2733	001022	000000	WCS2:	0
2734	001024	000000	WDS:	0
2735	001026	000000	WER:	0
2736				
2737				;CORE DUMP PATTERNS*****
2738				
2739	001030	000005	WCDP2:	5
2740	001032	000005		5
2741	001034	000012		12
2742	001036	000012		12
2743	001040	000000		0
2744	001042	000017	WCDPO:	17
2745	001044	000017		17
2746	001046	000017		17
2747	001050	000017		17
2748	001052	000000		0

2750
2751
2752
2753 001054 000000
2754 001056 000000
2755 001060 002636
2756 001062 002636
2757 001064 003120
2758 001066 003120
2759 001070 003324
2760 001072 003326
2761 001074 003506
2762 001076 003506
2763 001100 004006
2764 001102 004014
2765 001104 004176
2766 001106 004200
2767 001110 004312
2768 001112 004314
2769 001114 004426
2770 001116 004430
2771 001120 004552
2772 001122 004554
2773 001124 004776
2774 001126 005000
2775 001130 005172
2776 001132 005202
2777 001134 005274
2778 001136 005334
2779 001140 005410
2780 001142 005450
2781 001144 005524
2782 001146 005564
2783 001150 005640
2784 001152 005700
2785 001154 005754
2786 001156 005770
2787 001160 006116
2788 001162 006132
2789 001164 006262
2790 001166 006276
2791 001170 006406
2792 001172 006422
2793 001174 006536
2794 001176 006552
2795 001200 007062
2796 001202 007070
2797 001204 007214
2798 001206 007222
2799 001210 007430
2800 001212 007456
2801 001214 007550
2802 001216 007572
2803 001220 010066
2804 001222 010074
2805 001224 010700

;LOGIC TEST ENTRY TABLE*****

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

2806	001226	010714	LT32IT
2807	001230	011042	LT33
2808	001232	011056	LT33IT
2809	001234	011144	LT34
2810	001236	011160	LT34IT
2811	001240	011300	LT35
2812	001242	011314	LT35IT
2813	001244	011452	LT36
2814	001246	011466	LT36IT
2815	001250	011572	LT37
2816	001252	011606	LT37IT
2817	001254	011742	LT40
2818	001256	011756	LT40IT
2819	001260	012062	LT41
2820	001262	012076	LT41IT
2821	001264	012324	LT42
2822	001266	012362	LT42IT
2823	001270	012654	LT43
2824	001272	012702	LT43IT
2825	001274	013104	LT44
2826	001276	013132	LT44IT
2827	001300	013352	LT45
2828	001302	013400	LT45IT
2829	001304	013616	LT46
2830	001306	013644	LT46IT
2831	001310	014054	LT47
2832	001312	014070	LT47IT
2833	001314	014244	LT50
2834	001316	014260	LT50IT
2835	001320	014420	LT51
2836	001322	014434	LT51IT
2837	001324	002522	
2838	001326	000051	

TADX: .WORD TEND
TLAST: .WORD 51

:CONTAINS # OF TESTS

```

2840          .EVEN
2841          ;PROGRAM START AND HOUSEKEEPING*****
2842
2843          ;NOTE: PROGRAM STARTS HERE ON START AT 200
2844 001330 012706 000500          START: MOV #500,SP          ;SET STACK POINTER
2845 001334 013746 000004          MOV @#4,-(SP)          ;SAVE ERROR TRAP VECTOR
2846 001340 013746 000006          MOV @#5,-(SP)          ;AND VECTOR +2
2847 001344 012737 001370 000004          MOV #1$,@#4          ;SET NEW VECTOR
2848 001352 005037 000006          CLR @#6              ;AND PSW
2849 001356 022777 177777 177204          CMP #-1,@SWR          ;USE SOFTWARE SWITCH IF HARDWARE
2850 001364 001402          BEQ 2$              ;IS = 177777
2851 001366 000404          BR 3$              ;OTHERWISE USE HARDWARE SWR
2852 001370 022626          1$: CMP (SP)+,(SP)+          ;RESET STACK PTR
2853 001372 012737 000176 000570          2$: MOV #SWREG,SWR          ;SET SOFTWARE SWITCH REGISTER
2854 001400 012637 000006          3$: MOV (SP)+,@#6          ;RESTORE ERROR TRAP VECTORS
2855 001404 012637 000004          MOV (SP)+,@#4
2856 001410 005037 001014          CLR SKAT            ;CLEAR SKIP ADDRESS TEST FLAG
2857 001414 005027          CLR (PC)+          ;;CLEAR CHAIN INDICATOR
(1) 001416 000000          CHNFLG: .WORD 0          ;;CHAIN MODE INDICATOR
(1)          ;;1/0 = CHAIN/NOT CHAIN MODE
(1) 001420 022737 002566 000042          CMP #SENDAD,@#42          ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
(1) 001426 001404          BEQ 50$
(1) 001430 005737 000042          TST @#42              ;;BRANCH IF IN DUMP MODE
(1) 001434 001413          BEQ 52$
(1) 001436 000406          BR 51$
(1) 001440 012737 000176 000570          50$: MOV #SWREG,SWR          ;;INVOKE SOFTWARE SWR
(1) 001446 012777 100000 177114          MOV #100000,@SWR          ;;WITH HALT ON ERROR SET
(1) 001454 005237 001416          51$: INC CHNFLG          ;;SET CHNFLG = CHAIN MODE
(1) 001460 000137 002106          JMP TSCD              ;;GO TO CHAIN ADDRESS
(1) 001464          52$:
2858 001464 122737 000006 000041          4$: CMPB #6,@#41          ;BRANCH IF NOT LOADED VIA TMDP
2859 001472 001004          BNE 5$
2860 001474 012704 022750          MOV #MSG62,R4          ;ADVISE USER TO REMOVE TMDP FROM
2861 001500 004737 017446          JSR PC,TTOUT          ;UNIT UNDER TEST
2862 001504 012704 020364          5$: MOV #MSG1,R4
2863 001510 004737 017446          JSR PC,TTOUT          ;PRINT TITLE
2864 001514 112737 000043 020364          MOVB #'#,MSG1          ;DO NOT PRINT TITLE ON RESTART
2865 001522 012704 022320          MOV #MSG44,R4
2866 001526 004737 017446          JSR PC,TTOUT          ;REQUEST REGISTER ADDRESS
2867 001532 013703 000612          MOV REGS,R3
2868 001536 004737 017574          JSR PC,OCTP          ;PRINT CURRENT ADDRESS
2869 001542 012705 000612          MOV #REGS,R5          ;SET ADDRESS SAVE LOC
2870 001546 012701 000007          MOV #7,R1             ;SET SIZE OF RESPONSE
2871 001552 012702 176400          MOV #176400,R2        ;SET UPPER LIMIT
2872 001556 012703 172300          MOV #172300,R3        ;SET LOWER LIMIT
2873 001562 004737 017106          JSR PC,TTR           ;GO GET RESPONSE
2874 001566 012704 022342          MOV #MSG45,R4
2875 001572 004737 017446          JSR PC,TTOUT          ;REQUEST VECTOR
2876 001576 013703 000610          MOV VECT,R3
2877 001602 004737 017574          JSR PC,OCTP          ;PRINT CURRENT VECTOR
2878 001606 012705 000610          MOV #VECT,R5          ;SET ADDRESS SAVE LOC
2879 001612 012701 000004          MOV #4,R1             ;SET SIZE OF RESPONSE
2880 001616 012702 000224          MOV #224,R2           ;SET UPPER LIMIT
2881 001622 012703 000150          MOV #150,R3           ;SET LOWER LIMIT
2882 001626 004737 017106          JSR PC,TTR           ;GO GET RESPONSE
2883 001632 013700 000610          MOV VECT,R0           ;GET VECTOR

```

2884	001636	012720	016750	MOV	#MTINT,(R0)+	:LOAD INTERRUPT ADDRESS IN VECTOR
2885	001642	012710	000340	MOV	#340,(R0)	:LOAD PRIORITY
2886	001646	013700	000612	MOV	REGS,R0	:GET START OF REGS
2887	001652	012701	000016	MOV	#16,R1	:SET NUMBER OF REGS
2888	001656	012702	000510	MOV	#C1,R2	:GET START OF TABLE
2889	001662	010022		STO: MOV	R0,(R2)+	:BUILD TABLE
2890	001664	062700	000002	ADD	#2,R0	:BUMP ADDRESS
2891	001670	005301		DEC	R1	:SEE IF DONE
2892	001672	001373		BNE	STO	:IF NOT: BR
2893	001674	012702	000614	MOV	#TOB,R2	
2894	001700	012700	000077	MOV	#77,R0	
2895	001704	005022		ST1: CLR	(R2)+	:CLEAR FLAGS + COUNTERS
2896	001706	005300		DEC	R0	
2897	001710	001375		BNE	ST1	
2898	001712	012704	022710	MOV	#MSG57,R4	:REQUEST TM03 DRIVE #
2899	001716	004737	017446	JSR	PC,TTOUT	
2900	001722	013703	000624	MOV	DRVN,R3	:GET CURRENT DRIVE #
2901	001726	004737	017574	JSR	PC,OC1P	:PRINT IT
2902	001732	012705	000624	MOV	#DRVN,R5	:TTR ROUTINE RETURNS USER VALUE TO (R5)
2903	001736	012701	000002	MOV	#2,R1	:LIMIT RESPONSE
2904	001742	012702	000007	MOV	#7,R2	:LIMIT RANGE TO 0-7
2905	001746	012703	000000	MOV	#0,R3	
2906	001752	004737	017106	JSR	PC,1TR	:GET USER RESPONSE
2907	001756	012704	022726	MOV	#MSG58,R4	:REQUEST TU45 SLAVE #
2908	001762	004737	017446	JSR	PC,TTOUT	
2909	001766	013703	000664	MOV	SLVN,R3	:GET CURRENT SLAVE #
2910	001772	004737	017574	JSR	PC,OC1P	:AND PRINT IT
2911	001776	012705	000664	MOV	#SLVN,R5	:TTR ROUTINE RETURNS RESPONSE TO (R5)
2912	002002	012701	000002	MOV	#2,R1	:LIMIT RESPONSE TO 1 CHARACTER
2913	002006	012702	000007	MOV	#7,R2	:BETWEEN 0 AND 7
2914	002012	012703	000000	MOV	#0,R3	
2915	002016	004737	017106	JSR	PC,1TR	:GET USER RESPONSE
2916	002022	012704	022663	MOV	#MSG56,R4	
2917	002026	004737	017446	JSR	PC,TTOUT	:REQUEST STATIC ONLY
2918	002032	013703	001012	MOV	STATC,R3	:GET CURRENT VALUE
2919	002036	004737	017574	JSR	PC,OC1P	:AND TYPE IT
2920	002042	012705	001012	MOV	#STATC,R5	:SET ADDRESS OF STATIC FLAG
2921	002046	012701	000002	MOV	#2,R1	:SET SIZE OF RESPONSE
2922	002052	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
2923	002056	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
2924	002062	004737	017106	JSR	PC,1TR	:GET RESPONSE
2925						
2926				:START	210	
2927	002066	012706	000500	ST2: MOV	#500,SP	:SET STACK PTR
2928	002072	005037	001006	CLR	RDRVF	:CLEAR REVERSE FLAG
2929	002076	005037	001016	CLR	PCNTR	:CLEAR PASS COUNTER
2930	002102	004737	020222	JSR	PC,GTSWR	:GET SWITCHES

```

2932
2933 ;TEST SCHEDULAR*****
2934
2935 002106 052777 000100 176456 TSCD: BIS #100,@TKS ;SET KEYBOARD IE BIT
2936 002114 005037 001000 CLR WPGFL ;CLEAR WRAP PATRN FLAG
2937 002120 005037 000740 CLR STFLG ;CLEAR SINGLE TEST FLAG
2938 002124 017700 176440 MOV @SWR,R0
2939 002130 042700 177700 BIC #177700,R0 ;BRANCH IF SINGLE
2940 002134 001154 BNE STSCD ;TEST SELECTED
2941 002136 005737 001416 TST CHWFLG ;:BRANCH IF NOT IN CHAIN MODE
(1) 002142 001511 BEQ TSCDA
(1) 002144 012737 177777 000624 MOV #-1,DRVN ;:INITIALIZE DRIVE #
(1) 002152 012737 177777 000664 NXTDRV: MOV #-1,SLVN ;:INITIALIZE SLAVE #
(1) 002160 012777 000040 176332 1$: MOV #40,@CS ;:INIT CONTROLLER
(1) 002166 005237 000624 INC DRVN ;:STEP DRIVE #
(1) 002172 022737 000010 000624 CMP #10,DRVN ;:EXIT IF ALL DRIVES TESTED
(1) 002200 001553 BEQ $DONE ;:FOR AVAILABILITY
(1) 002202 013777 000624 176310 MOV DRVN,@CS ;:LOAD DRIVE #
(1) 002210 005777 176274 TST @C1 ;:ACCESS DRIVE
(1) 002214 032777 010000 176276 BIT #10000,@CS ;:BRANCH IF DRIVE NON EXISTANT
(1) 002222 001356 BNE 1$ ;:(NED = 1)
(1) 002224 005237 000664 NXTSLV: INC SLVN ;:STEP SLAVE # AND BRANCH
(1) 002230 001011 BNE 1$ ;:IF NOT SLAVE 0
(1) 002232 005737 000624 TST DRVN ;:BRANCH IF NOT DRIVE # 0
(1) 002236 001006 BNE 1$
(1) 002240 122737 000006 000041 CMPB #6,@#41 ;:BRANCH IF NOT TMDP
(1) 002246 001002 BNE 1$
(1) 002250 005237 000664 INC SLVN ;:STEP TO SLAVE # 1
(1) 002254 022737 000010 000664 1$: CMP #10,SLVN ;:BRANCH IF ALL SLAVES TESTED
(1) 002262 001733 BEQ NXTDRV ;:FOR AVAILABILITY
(1) 002264 013777 000664 176250 MOV SLVN,@C ;:LOAD SLAVE UNIT #
(1) 002272 032777 002000 176236 BIT #2000,@DT ;:BRANCH IF SLAVE NOT
(1) 002300 001751 BEQ NXTSLV ;:PRESENT (SPR = 0)
2942 002302 032777 010000 176212 BIT #10000,@DS ;TEST FOR MOL
2943 002310 001026 BNE TSCDA ;BRANCH IF MOL
2944 002312 012704 026543 MOV #MSG6,R4 ;ELSE PRINT NOT ON LINE
2945 002316 004737 017446 JSR PC,TTOUT ;PRINT MESSAGE
2946 002322 013703 000624 MOV DRVN,R3 ;DRIVE NUMBER
2947 002326 004737 017574 JSR PC,OCTP ;PRINT OCTAL NUMBER
2948 002332 012704 026555 MOV #MSG7,R4 ;
2949 002336 004737 017446 JSR PC,TTOUT ;PRINT MESSAGE
2950 002342 013703 000664 MOV SLVN,R3 ;SLAVE NUMBER
2951 002346 004737 017574 JSR PC,OCTP ;PRINT OCTAL
2952 002352 012704 026565 MOV #MSG8,R4 ;
2953 002356 004737 017446 JSR PC,TTOUT ;PRINT OCTAL
2954 002362 000137 002224 JMP NXTSLV
2955 002366 012737 001054 000742 TSCDA: MOV #TSTTBL,LTADD
2956 002374 062737 000004 000742 TSCD0: ADD #4,LTADD
2957 002402 013737 000742 000714 TSCD1: MOV LTADD,ITRLP
2958 002410 062737 000002 000714 ADD #2,ITRLP ;SEI ITERATION ADDRESS
2959 002416 005037 000620 CLR HDRFL ;CLEAR PRINT HEADER FLAG
2960 002422 017700 176314 MOV @LTADD,R0 ;SET POINTER TO TEST
2961 002426 000110 JMP (R0) ;GO TO TEST
2962 002430 032777 002000 176132 TSCD2: BIT #2000,@SWR ;SEE IF HALT ON TEST
2963 002436 001403 BEQ TSCD3 ;IF NOT: BR
2964 002440 000000 HALT

```

```

2965 002442 005037 001000
2966 002446 005737 000740
2967 002452 001750
2968 002454 017700 176110
2969 002460 042700 177700
2970 002464 001610
2971 002466 012737 000001 000740
2972 002474 023700 001326
2973 002500 002410
2974 002502 006300
2975 002504 006100
2976 002506 012737 001054 000742
2977 002514 060037 000742
2978 002520 000730
2979 002522 005737 001416
2980 002526 001236
2981 002530 012704 022160
2982 002534 004737 017446
2983 002540 013703 001016
2984 002544 004737 017574
2985 002550 005000
2986 002552 005300
2987 002554 001376
2988 002556 013700 000042
    (1) 002562 001405
    (1) 002564 000005
    (1) 002566 004710
    (1) 002570 000240
    (1) 002572 000240
    (1) 002574 000240
    (1) 002576 000240
2989 002600 005737 001416
2990 002604 001005
2991 002606 032777 010000 175754
2992 002614 001001
2993 002616 000000
2994 002620 012737 000001 001014
2995 002626 005237 001016
2996 002632 000137 002106

    CLR WPGFL ;CLEAR WRAP DATA GENERATOR FLAG
    TST STFLG ;SE IF SINGLE TEST
    BEQ TSCDO ;IF NOT: BR
    MOV @SWR,RO
    BIC #177700,RO ;BRANCH IF ALL TESTS DESIRED
    BEQ TSCD ;IF SO: BR
    MOV #1,STFLG ;SET SINGLE TEST FLAG
    CMP TLAST,RO ;SEE IF EXCEEDED TESTS
    BLT TEND ;IF SO: BR
    ASL RO
    ROL RO ;SET TABLE MODIFIER
    MOV #TSTTBL,LTADD
    ADD RO,LTADD ;SET TEST POINTER
    BR TSCD1
    TST CHNFLG ;BRANCH IF IN CHAIN MODE
    BNE NXTSLV ;STEP TO NEXT SLAVE
$DONE: MOV #MSG41,R4
    JSR PC,TTOUT ;PRINT END OF PASS
    MOV PCNTR,R3
    JSR PC,OCTP ;PRINT PASS NUMBER
    CLR RO ;DELAY WAITING FOR
    DEC RO
    BNE 1$
    MOV @#42,RO ;GET ACT11 RETURN ADDRESS
    BEQ HERE ;BRANCH IF NOT ACT11
    RESET
$ENDAD: JSR PC,(RO)
    NOP
    NOP
    NOP
    HERE: NOP
    TST CHNFLG ;BRANCH IF IN CHAIN MODE
    BNE TENDX
    BIT #10000,@SWR ;SEE IF HALT ON PASS
    BNE TENDX ;IF NOT: BR
    HALT
    TENDX: MOV #1,SKAT ;SET SKIP ADDRESS TEST FLAG
    INC PCNTR ;BUMP PASS COUNTER
    JMP TSCD ;RESTART
    
```



```

2998 ;LOGIC TEST 1: DRIVE ADDRESSING*****
2999
3000 002636 013737 000624 000700 LT1:  MOV   DRVN,TEMP3  ;GET DRIVE # TO BE TESTED
3001 002644 013701 000624          MOV   DRVN,R1
3002 002650 012737 023044 000622  MOV   #MSLT1,EMADDR
3003 002656 005737 001014          TST   SKAT          ;SEE IF SKIP ADDRESS TESTS
3004 002662 001403          BEQ   1$           ;IF NOT: BR
3005 002664 005737 000740          TST   STFLG        ;SEE IF SINGLE TEST
3006 002670 001511          BEQ   LT1X         ;IF NOT: BR
3007 002672 032777 001000 175670 1$:  BIT   #1000,@SWR   ;BRANCH IF MAN INTERVENTION
3008 002700 001433          BEQ   LT1A         ;NOT SELECTED
3009 002702 012704 020623          LT1G0: MOV   #MSG2,R4
3010 002706 004737 016702          JSR   PC,INST      ;PRINT TEST INSTRUCTIONS
3011 002712 012737 023044 000622 LT1G:  MOV   #MSLT1,EMADDR ;SET HEADFR ADDRESS
3012 002720 012704 020562          MOV   #MSG2,R4
3013 002724 004737 017446          JSR   PC,TTOUT     ;REQUEST DRIVE NUMBER
3014 002730 012705 000700          MOV   #TEMP3,R5   ;TTR ROUTINE RETURNS RESPONSE TO (R5)
3015 002734 012701 000002          MOV   #2,R1
3016 002740 012702 000007          MOV   #7,R2
3017 002744 012703 000000          MOV   #0,R3
3018 002750 004737 017106          JSR   PC,TTR       ;GET DRIVE NUMBER
3019 002754 005737 000674          TST   TEMP1        ;SEE IF ANOTHER DRIVE
3020 002760 001455          BEQ   LT1X         ;IF NOT: BR
3021 002762 005001          CLR   R1           ;SELECT DRIVE 0
3022 002764 012700 000010          MOV   #10,R0       ;SET NUMBER OF DRIVES
3023 002770 012777 000040 175522 LT1A:  MOV   #40,@CS      ;INIT
3024 002776 010177 175516          MOV   R1,@CS       ;SELECT DRIVE
3025 003002 005777 175502          TST   @C1          ;ACCESS DRIVE
3026 003006 032777 010000 175504  BIT   #10000,@CS   ;SEE IF NED
3027 003014 001010          BNE   LT1B         ;IF SO: BR
3028 003016 032777 001000 175544  BIT   #1000,@SWR   ;BRANCH IF NOT MANUAL INTERVENTION
3029 003024 001433          BEQ   LT1X
3030 003026 023701 000700          CMP   TEMP3,R1     ;SEE IF SHOULD BE NED
3031 003032 001404          BEQ   LT1C         ;IF NOT: BR
3032 003034 000407          BR    LT1ER        ;ELSE GO TO ERROR
3033 003036 023701 000700          LT1B:  CMP   TEMP3,R1 ;SEE IF SHOULD BE NED
3034 003042 001410          BEQ   LT1ER1
3035 003044 005300          LT1C:  DEC   R0
3036 003046 001721          BEQ   LT1G         ;IF DONE ALL: BR
3037 003050 005201          INC   R1           ;SELECT NEXT DRIVE
3038 003052 000746          BR    LT1A         ;CONTINUE
3039 003054 012737 000001 000716 LT1ER:  MOV   #1,EXFL      ;FLAG EXPT
3040 003062 000403          BR    LT1ER2
3041 003064 012737 000002 000716 LT1ER1: MOV   #2,EXFL      ;FLAG NOT EXPT
3042 003072 012737 020752 000672 LT1ER2: MOV   #MSG3,ERADD ;FLAG CONDITION
3043 003100 012737 002770 000712  MOV   #LT1A,SCOLP  ;SET SCOPE ADDRESS
3044 003106 004737 015000          JSR   PC,LTGER     ;GO PRINT LOGIC TEST ERROR
3045 003112 000754          BR    LT1C         ;CONTINUE TEST
3046 003114 000137 002430          LT1X:  JMP   TSCD2      ;RETURN TO SCHED
3047

```

```

3049 ;LOGIC TEST 2: REGISTER ADDRESSING*****
3050
3051 003120 000240 LT2: NOP
3052 003122 012777 000040 175370 LT2IT: MOV #40,@CS ;INIT
3053 003130 013777 000624 175362 MOV DRVN,@CS ;SELECT DRIVE
3054 003136 012737 023120 000622 #MSLT2,EMADDR ;SAVE LT2 HEADER ADDRESS
3055 003144 012705 000510 MOV #C1,R5 ;SET ADDRESS OF FIRST REGISTER
3056 003150 012700 000016 MOV #16,R0 ;SET NUMBER OF REGISTERS
3057 003154 012702 000626 MOV #TR00,R2 ;SET START OF REGISTER BUFFER
3058 003160 011501 LT2A: MOV (R5),R1
3059 003162 011112 MOV (R1),(R2) ;READ REGISTER
3060 003164 032777 020000 175316 BIT #20000,@C1 ;SEE IF ERROR
3061 003172 001402 BEQ LT2B ;IF NOT: BR
3062 003174 004737 003224 JSR PC,LT2ER1 ;ELSE GO TO ERROR 1
3063 003200 032777 000002 175316 LT2B: BIT #2,@ER ;SEE IF ILR
3064 003206 001402 BEQ LT2C ;IF NOT: BR
3065 003210 004737 003242 JSR PC,LT2ER2 ;ELSE GO TO ERROR 2
3066 003214 022225 LT2C: CMP (R2)+,(R5)+ ;BUMP ADDRESS
3067 003216 005300 DEC R0
3068 003220 001357 BNE LT2A ;CONTINUE FOR ALL REGISTERS
3069 003222 000434 BR LT2X
3070 003224 012737 000002 000716 LT2ER1: MOV #2,EXFL ;FLAG NOT EXPECTED
3071 003232 012737 020774 000672 MOV #MSG4,ERADD ;POINT TO CONTROLLER ERROR
3072 003240 000415 BR LT2ERG ;GO TO ERROR
3073 003242 012737 000002 000716 LT2ER2: MOV #2,EXFL ;FLAG NOT EXPECTED
3074 003250 012737 021012 000672 MOV #MSG5,ERADD ;POINT TO DRIVE ERROR
3075 003256 000406 BR LT2ERG ;GO TO ERROR
3076 003260 012737 000001 000716 LT2ER3: MOV #1,EXFL ;FLAG EXPECTED
3077 003266 012737 020774 000672 MOV #MSG4,ERADD ;POINT TO DRIVE
3078 003274 012737 003310 000712 LT2ERG: MOV #LT2LP,SCOLP ;SET SCOPE ADDRESS
3079 003302 004737 015000 JSR PC,LTGER ;GO PRINT
3080 003306 000207 RTS PC ;ELSE CONTINUE
3081 003310 005726 LT2LP: TST (SP)+ ;RESET STACK
3082 003312 000722 BR LT2A ;LOOP
3083 003314 004737 016350 LT2X: JSR PC,ITER ;GO SEE IF ITERATIONS
3084 003320 000137 002430 JMP TSCD2 ;RETURN TO SCHED
  
```

```

3086                                     ;LOGIC TEST 3: CONTROL BUS*****
3087
3088 003324 000240 LT3: NOP
3089 003326 012737 023177 000622 LT3IT: MOV #MSLT3,EMADDR ;SET TEST HEADER
3090 003334 012701 000001 MOV #1,R1 ;PRESET PATTERN 1
3091 003340 012700 000020 MOV #20,R0 ;SET PATTERN CHANGE NUMBER
3092 003344 004737 016440 LT3A: JSR PC,INIT1 ;GO INIT
3093 003350 010177 175142 MOV R1,@FC ;WRITE TO FC
3094 003354 032777 000010 175142 BIT #10,@ER ;SEE IF CPAR (TM03)
3095 003362 001012 LT3ER1 ;IF SO: BR
3096 003364 017702 175126 LT3B: MOV @FC,R2 ;READ FC
3097 003370 032777 020000 175112 BIT #20000,@C1 ;SEE IF MCPE (RH)
3098 003376 001017 BNE LT3ER2 ;IF SO: BR
3099 003400 005300 LT3C: DEC R0 ;SEE IF DONE PATTERN CHANGES
3100 003402 001426 BEQ LT3X ;IF SO: BR
3101 003404 006301 ASL R1 ;CHANGE PATTERN
3102 003406 000756 BR LT3A ;CONTINUE
3103 003410 012737 021325 000672 LT3ER1: MOV #MSG11,ERADD ;SET ERROR CODE
3104 003416 012737 003344 000712 MOV #LT3A,SCOLP ;SET SCOPE ADDRESS
3105 003424 017702 175066 MOV @FC,R2 ;GET DATA
3106 003430 004737 016106 JSR PC,LTGER1 ;GO DO ERROR
3107 003434 000753 BR LT3B
3108 003436 012737 021301 000672 LT3ER2: MOV #MSG10,ERADD ;SET ERROR CODE
3109 003444 012737 003364 000712 MOV #LT3B,SCOLP ;SET SCOPE ADDRESS
3110 003452 004737 016106 JSR PC,LTGER1 ;GO DO ERROR
3111 003456 000750 BR LT3C
3112 003460 105701 LT3X: TSTB R1 ;SEE IF DONE PATTERN 2
3113 003462 100405 BMI LT3XX ;IF SO: BR
3114 003464 012701 000401 MOV #401,R1 ;SET PATTERN 2
3115 003470 012700 000010 MOV #10,R0 ;SET PATTERN CHANGE NUMBER
3116 003474 000723 BR LT3A ;DO PATTERN 2
3117 003476 004737 016350 LT3XX: JSR PC,ITER ;GO SEE IF ITERATIONS
3118 003502 000137 002430 JMP TSCD2 ;RETURN TO SCHEDULAR
  
```

```

3120
3121          ;LOGIC TEST 4: SLAVE ADDRESSING*****
3122
3123 003506 013737 000664 000700 LT4:  MOV    SLVN,TEMP3
3124 003514 013701 000664          MOV    SLVN,R1
3125 003520 005737 001014          TST    SKAT          ;SEE IF SKIP ADDRESS TESTS
3126 003524 001403          BEQ    1$           ;IF NOT: BR
3127 003526 005737 000740          TST    STFLG        ;SEE IF SINGLE TEST
3128 003532 001523          BEQ    LT4X         ;IF NOT: BR
3129 003534 032777 001000 175026 1$:  BIT    #1000,@SWR   ;BRANCH IF MAN INTERVENTION
3130 003542 001430          BEQ    LT4A         ;NOT SELECTED
3131 003544 012704 021130          LT4G0: MOV    #MSG8A,R4
3132 003550 004737 016702          JSR    PC,INST      ;PRINT TEST INSTRUCTIONS
3133 003554 012704 021067          LT4G:  MOV    #MSG8,R4
3134 003560 004737 017446          JSR    PC,TTOUT     ;REQUEST SLAVE
3135 003564 012705 000700          MOV    #TEMP3,R5
3136 003570 012701 000002          MOV    #2,R1
3137 003574 012702 000007          MOV    #7,R2
3138 003600 012703 000000          MOV    #0,R3
3139 003604 004737 017106          JSR    PC,ITR       ;GET SLAVE NUMBER
3140 003610 005737 000674          TST    TEMP1        ;SEE IF SLAVE
3141 003614 001472          BEQ    LT4X         ;IF NOT: BR
3142 003616 005001          CLR    R1           ;SELECT SLAVE 0
3143 003620 012700 000010          MOV    #10,R0       ;SET NUMBER OF SLAVES
3144 003624 012777 000040 174666 LT4A: MOV    #40,@CS     ;INIT
3145 003632 013777 000624 174660          MOV    DRVN,@CS    ;SELECT DRIVE
3146 003640 010177 174676          MOV    R1,@TC       ;SELECT SLAVE
3147 003644 017703 174666          MOV    @DT,R3       ;GET DT
3148 003650 020137 000700          CMP    R1,TEMP3     ;SEE IF SHOULD HAVE SPR
3149 003654 001404          BEQ    LT4B         ;IF SO: BR
3150 003656 032703 002000          BIT    #2000,R3    ;SEE IF SPR
3151 003662 001420          BEQ    LT4D         ;IF NOT: BR
3152 003664 000423          BR     LT4ER1       ;GO TO ERROR 1
3153 003666 032703 002000          LT4B:  BIT    #2000,R3 ;SEE IF NO SLAVE PRESENT
3154 003672 001424          BEQ    LT4ER2       ;(SPR=0)
3155 003674 012704 022002          LT4C:  MOV    #MSG30,R4
3156 003700 004737 017446          JSR    PC,TTOUT     ;PRINT SERIAL NUMBER TAG
3157 003704 017703 174630          MOV    @SN,R3
3158 003710 004737 020120          JSR    PC,SNPT      ;PRINT SERIAL NUMBER
3159 003714 032777 001000 174646          BIT    #1000,@SWR  ;BRANCH IF NOT MANUAL INTERVENTION
3160 003722 001427          BEQ    LT4X
3161 003724 005300          LT4D:  DEC    R0
3162 003726 001712          BEQ    LT4G         ;IF DONE ALL: BR
3163 003730 005201          INC    R1           ;BUMP SLAVE
3164 003732 000734          BR     LT4A         ;CONTINUE
3165 003734 012737 000001 000716 LT4ER1: MOV    #1,EXFL    ;FLAG EXPT: NOT RECEIVED
3166 003742 000403          BR     LT4ERG
3167 003744 012737 000002 000716 LT4ER2: MOV    #2,EXFL    ;FLAG RECVD: NOT EXPT
3168 003752 012737 023264 000622 LT4ERG: MOV    #MSLT4,EMADDR ;SET LT4 HEADER
3169 003760 012737 021257 000672          MOV    #MSG9,ERADD ;SET ERROR CONDITION
3170 003766 012737 003624 000712          MOV    #LT4A,SCOLP ;SET SCOPE ADDRESS
3171 003774 004737 015000          JSR    PC,LTGER     ;GO TO ERROR
3172 004000 000751          BR     LT4D         ;IF NO SCOPE: BR
3173 004002 000137 002430          LT4X:  JMP    TSCD2    ;RETURN TO SCHED
3174

```

```

3176                                     ;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST*****
3177
3178 004006 012737 023346 000622 LT5:  MOV  #MSLT5,EMADDR  ;SET TEST HEADER
3179 004014 004737 016440          LT5IT: JSR  PC,INIT1    ;GO INIT
3180 004020 012700 000032          MOV  #32,R0        ;SET LOOP FOR BITS 4-0
3181 004024 005001          CLR  R1           ;SET TEST WORD
3182 004026 010177 174502          LT5A: MOV  @MR,R1    ;SEND TEST WORD TO MR
3183 004032 017702 174476          MOV  @MR,R2       ;READ MR
3184 004036 042702 177740          BIC  #177740,R2   ;MASK BITS 4-0
3185 004042 020102          CMP  R1,R2        ;SEE IF EXPT = RECVD
3186 004044 001026          BNE  LT5ER1
3187 004046 005300          LT5B: DEC  R0
3188 004050 001402          BEQ  LT5C         ;IF DONE LOOP: BR
3189 004052 005201          INC  R1           ;BUMP TEST WORD
3190 004054 000764          BR   LT5A        ;CONTINUE LOOP
3191 004056 012701 000015          LT5C: MOV  #15,R1   ;SET TEST WORD + WAM 3
3192 004062 012700 001000          MOV  #1000,R0    ;SET LOOP FOR BITS 15-7
3193 004066 010177 174442          LT5D: MOV  @MR,R1   ;LOAD MR
3194 004072 017702 174436          MOV  @MR,R2       ;READ MR
3195 004076 042702 000140          BIC  #140,R2     ;MASK OUT BITS 5,6
3196 004102 020102          CMP  R1,R2        ;SEE IF EXPT = RECVD
3197 004104 001401          BEQ  LT5E        ;IF SO: BR
3198 004106 000416          BR   LT5ER2      ;ELSE GO TO ERR 2
3199 004110 005300          LT5E: DEC  R0
3200 004112 001425          BEQ  LT5X        ;IF DONE LOOP: BR
3201 004114 062701 000200          ADD  #200,R1     ;BUMP TEST WORD
3202 004120 000762          BR   LT5D        ;CONTINUE LOOP
3203 004122 012737 021370 000672          LT5ER1: MOV #MSG14,ERADD ;SET ERROR CODE
3204 004130 012737 004026 000712          MOV  #LT5A,SCOLP ;SET SCOPE ADDRESS
3205 004136 004737 016106          JSR  PC,LTGER1   ;GO TO ERROR
3206 004142 000741          BR   LT5B        ;CONTINUE
3207 004144 012737 021405 000672          LT5ER2: MOV #MSG15,ERADD ;SET ERROR CODE
3208 004152 012737 004066 000712          MOV  #LT5D,SCOLP ;SET SCOPE ADDRESS
3209 004160 004737 016106          JSR  PC,LTGER1   ;GO TO ERROR
3210 004164 000751          BR   LT5E        ;CONTINUE
3211 004166 004737 016350          LT5X: JSR  PC,ITER ;GO SEE IF ITERATIONS
3212 004172 000137 002430          JMP  TSCD2       ;RETURN TO SCHED
3213

```

```

3215                                     ;LOGIC TEST 6: TC REGISTER BIT TEST*****
3216
3217 004176 000240
3218 004200 012737 023415 000622 LT6:  NOP
3219 004206 012700 000003 LT6IT: MOV #MSLT6,EMADDR ;POINT TO LT6 HEADER
3220 004212 005001 LT6A1: MOV #3,R0 ;SET NUMBER OF TESTS
3221 004214 004737 016440 LT6A: JSR PC,INIT1 ;GO INIT
3222 004220 010177 174316 LT6B: MOV R1,@TC ;WRITE TC
3223 004224 017702 174312 MOV @TC,R2 ;READ TC
3224 004230 042702 160000 BIC #160000,R2 ;MASK OUT SAC
3225 004234 020102 CMP R1,R2 ;SEE IF EXPT = RECDV
3226 004236 001010 BNE LT6ER1 ;IF NOT: BR
3227 004240 005300 LT6D: DEC R0
3228 004242 001417 BEQ LT6X ;IF DONE ALL: BR
3229 004244 022700 000001 CMP #1,R0 ;SEE IF RESET TEST
3230 004250 001760 BEQ LT6A1 ;IF SO: BR
3231 004252 012701 017777 MOV #17777,R1 ;SET TEST WORD
3232 004256 000756 BR LT6A ;DO SET TEST
3233 004260 012737 021433 000672 LT6ER1: MOV #MSG18,ERADD ;SET ERROR CODE
3234 004266 012737 004220 000712 MOV #LT6B,SCOLP ;SET SCOPE ADDRESS
3235 004274 004737 016106 JSR PC,LTGER1 ;GO TO ERROR
3236 004300 000757 BR LT6D ;CONTINUE
3237 004302 004737 016350 LT6X: JSR PC,ITER ;GO SEE IF ITERATIONS
3238 004306 000137 002430 JMP TSCD2 ;RETURN TO SCHED
3239
  
```

```

3241                                     ;LOGIC TEST 7: FRAME COUNT BIT TEST*****
3242
3243 004312 000240                       LT7:  NOP
3244 004314 012700 0C0003                LT7IT: MOV  #3,R0           ;SET TEST NUMBER
3245 004320 012737 023464 000622        LT7C:  MOV  #MSLT7,EMADDR ;SET TEST HEADER
3246 004326 005001                       CLR  R1                   ;SET TEST WORD
3247 004330 004737 016440                LT7A:  JSR  PC,INIT1      ;GO INIT
3248 004334 010177 174156                MOV  R1,@FC              ;CLEAR FRAME COUNT
3249 004340 017702 174152                MOV  @FC,R2              ;READ FC
3250 004344 020102                       CMP  R1,R2               ;SEE IF EXPT = RECVD
3251 004346 001010                       BNE  LT7ER1
3252 004350 005300                       LT7B:  DEC  R0            ;SEE IF DONE ALL
3253 004352 001417                       BEQ  LT7X                ;IF SO: BR
3254 004354 022700 000001                CMP  #1,R0               ;SEE IF RESET TEST
3255 004360 001757                       BEQ  LT7C                ;IF SO: BR
3256 004362 012701 177777                MOV  #-1,R1              ;SET TEST WORD TO -1
3257 004366 000760                       BR   LT7A                ;CONTINUE
3258 004370 012737 021452 000672        LT7ER1: MOV #MSG19,ERADD ;SET ERROR CODE
3259 004376 012737 004330 000712        MOV  #LT7A,SCOLP        ;SET SCOPE ADDRESS
3260 004404 004737 016106                JSR  PC,LTGER1          ;GO PRINT ERROR
3261 004410 000757                       BR   LT7B                ;ELSE CONTINUE
3262 004412 012700 000003                LT7X:  MOV  #3,R0        ;RESET TEST AMT
3263 004416 004737 016350                JSR  PC,ITER            ;GO SEE IF ITERATIONS
3264 004422 000137 002430                JMP  TSCD2              ;RETURN TO SCHED
3265
  
```

```

3267                                     :LOGIC TEST 10: FUNCTION CODE BIT TEST*****
3268
3269 004426 000240                      LT10:  NOP
3270 004430 012737 023533 000622      LT10IT: MOV    #MSLT10,EMADDR  :SET TEST HEADER
3271 004436 012700 000003              MOV    #3,R0                :SET NUMBER OF TESTS
3272 004442 005001                      LT10A1: CLR    R1            :SET TEST WORDD
3273 004444 012777 000040 174046      LT10A:  MOV    #40,@CS       :INIT
3274 004452 013777 000624 174040      MOV    DRVN,@CS            :SELECT DRIVE
3275 004460 010177 174024              MOV    R1,@C1              :WRITE C1
3276 004464 017702 174020              MOV    @C1,R2              :READ C1
3277 004470 042702 177701              BIC    #177701,R2          :MASK FUNCTION CODE
3278 004474 020102                      CMP    R1,R2                :SEE IF EXPT = RECVD
3279 004476 001010                      BNE    LT10E1
3280 004500 005300                      LT10B:  DEC    R0
3281 004502 001417                      BEQ    LT10X                :IF DONE ALL: BR
3282 004504 022700 000001              CMP    #1,R0                :SEE IF RESET TEST
3283 004510 001754                      BEQ    LT10A1              :IF SO: BR
3284 004512 012701 000076              MOV    #76,R1              :SET TEST WORD
3285 004516 000752                      BR     LT10A                :DO SET TEST
3286 004520 012737 021471 000672      LT10E1: MOV    #MSG20,ERADD    :SET ERROR CODE
3287 004526 012737 004444 000712      MOV    #LT10A,SCOLP        :SET SCOPE ADDRESS
3288 004534 004737 016106              JSR    PC,LTGER1           :GO PRINT ERROR
3289 004540 000757                      BR     LT10B                :ELSE CONTINUE
3290 004542 004737 016350              LT10X:  JSR    PC,ITER       :GO SEE IF ITERATIONS
3291 004546 000137 002430              JMP    TSCD2               :RETURN TO SCHED
  
```



```

3293
3294
3295
3296 004552 000240
3297 004554 012737 023611 000622
3298 004562 004737 016440
3299 004566 017702 173716
3300 004572 032702 000001
3301 004576 001030
3302 004600 012777 000015 173726
3303 004606 005077 173704
3304 004612 052777 001700 173722
3305 004620 012777 000071 173662
3306 004626 017702 173656
3307 004632 032702 000001
3308 004636 001424
3309 004640 004737 016440
3310 004644 017702 173640
3311 004650 032702 000001
3312 004654 001444
3313 004656 000430
3314 004660 012737 021523 000672
3315 004666 012702 000001
3316 004672 005001
3317 004674 012737 004554 000712
3318 004702 004737 016106
3319 004706 000734
3320 004710 012737 021561 000672
3321 004716 005002
3322 004720 012701 000001
3323 004724 012737 004600 000712
3324 004732 004737 016106
3325 004736 000740
3326 004740 012737 021602 000672
3327 004746 005001
3328 004750 012702 000001
3329 004754 012737 004640 000712
3330 004762 004737 016106
3331 004766 004737 016350
3332 004772 000137 002430

;LOGIC TEST 11: GO BIT SET RESET*****
LT11: NOP
LT11IT: MOV #MMLT11,EMADDR ;SET TEST HEADER
JSR PC,INIT1 ;GO INIT
MOV @C1,R2 ;READ C1
BIT #1,R2 ;SEE IF GO=0
BNE LT11E1
LT11B: MOV #15,@MR ;SELECT WAM 3
CLR @FC ;ASSURE FCS = 1
BIS #1700,@TC ;ASSURE FMT OK
MOV #71,@C1 ;SET READ+GO
MOV @C1,R2 ;READ C1
BIT #1,R2 ;SEE IF GO =1
BEQ LT11E2
LT11C: JSR PC,INIT1 ;GO INIT
MOV @C1,R2 ;READ C1
BIT #1,R2 ;SEE IF GO=0
BEQ LT11X ;IF SO:BR
BR LT11E3 ;ELSE GO TO ERROR 3
LT11E1: MOV #MSG21,ERADD ;SET ERROR CODE
MOV #1,R2 ;SET REVD
CLR R1 ;SET EXPT
;SET SCOPE ADDRESS
MOV #LT11IT,SCOLP
JSR PC,LTGER1 ;GO PRINT ERROR
BR LT11B ;ELSE CONTINUE
LT11E2: MOV #MSG22,ERADD ;SET ERROR CODE
CLR R2 ;SET RCVD
MOV #1,R1 ;SET EXPT
;SET SCOPE ADDRESS
MOV #LT11B,SCOLP
JSR PC,LTGER1 ;GO PRINT ERROR
BR LT11C ;ELSE CONTINUE
LT11E3: MOV #MSG23,ERADD ;SET ERROR CODE
CLR R1 ;SET EXPT
MOV #1,R2 ;SET RCVD
;SET SCOPE ADDRESS
MOV #LT11C,SCOLP
JSR PC,LTGER1 ;GO PRINT ERROR
LT11X: JSR PC,ITER ;GO SEE IF ITERATIONS
JMP TSCD2 ;RETURN TO SCHED
  
```

```

3334
3335                ;LOGIC TEST 12: DRIVE READY BIT*****
3336
3337 004776 000240      LT12:  NOP
3338 005000 012737 023656 000622  LT12IT: MOV    #MSLT12,EMADDR ;SET TEST HEADER
3339 005006 004737 016440          JSR    PC,INIT1   ;GO INIT
3340 005012 032777 000200 173502  BIT    #200,ADS   ;SEE IF DRY=1
3341 005020 001426          BEQ    LT12E1
3342 005022 012777 000015 173504  LT12B: MOV    #15,AMR   ;SET WAMB3
3343 005030 005077 173462          CLR    @FC        ;ASSURE FCS = 1
3344 005034 052777 001700 173500  BIS    #1700,@TC  ;ASSURE FMT OK
3345 005042 012777 000071 173440  MOV    #71,@C1   ;SET READ+GO
3346 005050 032777 000200 173444  BIT    #200,ADS   ;SEE IF DRY=0
3347 005056 001020          BNE    LT12E2
3348 005060 004737 016440          JSR    PC,INIT1   ;GO INIT
3349 005064 032777 000200 173430  BIT    #200,ADS   ;SEE IF DRY=1
3350 005072 001033          BNE    LT12X
3351 005074 000422          BR     LT12E3     ;IF SO: BR
3352 005076 012737 021635 000672  LT12E1: MOV    #MSG24,ERADD ;ELSE GO TO ERROR 3
3353 005104 012737 005000 000712  MOV    #LT12IT,SCOLP ;SET ERROR CODE
3354 005112 004737 016100          JSR    PC,LTGER2  ;SET SCOPE ADDRESS
3355 005116 000741          BR     LT12B     ;GO TO ERROR
3356 005120 012737 021663 000672  LT12E2: MOV    #MSG25,ERADD ;CONTINUE
3357 005126 012737 005022 000712  MOV    #LT12B,SCOLP ;SET ERROR CODE
3358 005134 004737 016100          JSR    PC,LTGER2  ;SET LOOP ADDRESS
3359 005140 000747          BR     LT12C     ;GO PRINT ERROR
3360 005142 012737 021712 000672  LT12E3: MOV    #MSG25A,ERADD ;CONTINUE
3361 005150 012737 005060 000712  MOV    #LT12C,SCOLP ;SET ERROR CODE
3362 005156 004737 016100          JSR    PC,LTGER2  ;SET ERROR LOOP
3363 005162 004737 016350          JSR    PC,ITER   ;GET PRINT ERROR
3364 005166 000137 002430          JMP    TSCD2     ;GO TO ITERATION SUBROUTINE
                    ;RETURN TO SCHED

```

```
3366  
3367  
3368  
3369 005172 005000  
3370 005174 012737 023727 000622  
3371 005202 004737 016440  
3372 005206 012737 005264 000670  
3373 005214 005077 173270  
3374 005220 005077 173342  
3375 005224 052777 000100 173256  
3376 005232 005300  
3377 005234 001376  
3378 005236 012777 000340 173322  
3379 005244 012737 021737 000672  
3380 005252 012737 005202 000712  
3381 005260 004737 016100  
3382 005264 004737 016350  
3383 005270 000137 002430
```

:LOGIC TEST 13: INTERRUPT TEST*****

```
LT13: CLR R0  
MOV #MSLT13,EMADDR ;SET TEST HEADER  
LT13IT: JSR PC,INIT1 ;GO INIT,SELECT DRIVE, SELECT ABOVE  
MOV #LT13X,RTRN ;SET RETURN ADDRESS  
CLR @C1 ;CLEAR C51  
CLR @PSW ;SET PRIORITY  
BIS #100,@C1 ;BIT SET IE  
LT13A: DEC R0  
BNE LT13A ;AWAIT INTERRUPT  
LT13E1: MOV #340,@PSW ;RESET PRIORITY  
MOV #MSG26,ERADD ;SET ERROR CODE  
MOV #LT13IT,SCOLP ;SET LOOP ADDRESS  
JSR PC,LTGER2 ;GO PRINT ERROR  
LT13X: JSR PC,ITER ;GO TO ITERATION SUBROUTINE  
JMP TSCD2 ;RETURN TO SCHED
```

3385
 3386
 3387
 3388
 3389
 3390
 3391
 3392
 3393
 3394
 3395
 3396
 3397
 3398
 3399
 3400
 3401
 3402
 3403
 3404
 3405
 3406
 3407
 3408
 3409
 3410
 3411

005274 032777 001000 173266
 005302 001005
 005304 005737 000740
 005310 001435
 005312 000137 016420
 005316 012737 023774 000622
 005324 012704 026062
 005330 004737 016702
 005334 004737 016440
 005340 012701 014602
 005344 017702 173152
 005350 042702 000040
 005354 020102
 005356 001410
 005360 012737 005334 000712
 005366 012737 021766 000672
 005374 004737 016106
 005400 004737 016350
 005404 000137 002430

```

: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: 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
      BIC #40,R2 ;CLEAR PE, IF UP
      CMP R1,R2
      BEQ LT14X ;IF SO: BR
      MOV #LT14IT,SCOLP ;SET LOOP ADDRESS
      MOV #MSG27,ERADD ;SET ERROR CODE
      JSR PC,LTGER1 ;GO PRINT ERROR
LT14X: JSR PC,ITER ;GO SEE IF ITERATION
LT14XX: JMP TSCD2 ;RETURN TO SCHED
  
```

```

3413
3414           ;LOGIC TEST 15: STATUS AT BOT, OFFLINE, LOADED, NO WRITE RING*****
3415
3416 005410 032777 001000 173152 LT15: BIT #1000,@SWR ;SEE IF INHIB MAN TST
3417 005416 001005          BNE LT15A ;IF NOT: BR
3418 005420 005737 000740          TST STFLG ;SEE IF SINGLE TEST
3419 005424 001435          BEQ LT15XX ;IF NOT: BR
3420 005426 000137 016420          JMP INMT ;ELSE GO PRINT INHIB MSG
3421 005432 012737 024042 000622 LT15A: MOV #MSLT15,EMADDR ;SET TEST HEADER
3422 005440 012704 026160          MOV #MMMSG2,R4
3423 005444 004737 016702          JSR PC,INST ;PRINT INSTRUCTION
3424 005450 004737 016450          LT15IT: JSR PC,INIT2 ;GO INIT, SELECT DRIVE, SLAV
3425 005454 012701 100700          MOV #100700,R1 ;SET TEST WORD
3426 005460 017702 173036          MOV @DS,R2 ;READ STATUS
3427 005464 042702 000040          BIC #40,R2 ;CLEAR PE, IF UP
3428 005470 020102          CMP R1,R2 ;SEE OF EXPT=RCVD
3429 005472 001410          BEQ LT15X
3430 005474 012737 005450 000712          MOV #LT15IT,SCOLP ;SET LOOP ADDRESS
3431 005502 012737 021766 000672          MOV #MSG27,ERADD ;SET ERROR CODE
3432 005510 004737 016106          JSR PC,LTGER1 ;GO PRINT ERROR
3433 005514 004737 016350          LT15X: JSR PC,ITER ;GO SEE IF ITERATIONS
3434 005520 000137 002430          LT15XX: JMP TSCD2 ;RETURN TO SCHED
  
```

```
3436  
3437 ;LOGIC TEST 16: STATUS AT EOT, OFFLINE LOADED, NO WRITE RING*****  
3438  
3439 005524 032777 001000 173036 LT16: BIT #1000,@SWR :SEE IF INHIB MAN TST  
3440 005532 001005 BNE LT16A :IF NOT: BR  
3441 005534 005737 000740 TST STFLG :SEE IF SINGLE TEST  
3442 005540 001435 BEQ LT16XX :IF NOT: BR  
3443 005542 000137 016420 JMP INMT :ELSE GO PRINT INHIB MSG  
3444 005546 012737 024110 000622 LT16A: MOV #MSLT16,EMADDR :SET TEST HEADER  
3445 005554 012704 026201 MOV #MSG3,R4  
3446 005560 004737 016702 JSR PC,INST :GO PRINT INSTRUCTION  
3447 005564 004737 016450 LT16IT: JSR PC,INIT2 :SELECT DRIVE,SLAVE  
3448 005570 012701 116701 MOV #116701,R1 :SET TEST WORD  
3449 005574 017702 172722 MOV @DS,R2 :READ STATUS  
3450 005600 042702 000040 BIC #40,R2 :CLEAR PE, IF UP  
3451 005604 020102 CMP R1,R2 :SEE IF EXPT=RCVD  
3452 005606 001410 BEQ LT16X :IF SO: BR  
3453 005610 012737 005564 000712 MOV #LT16IT,SCOLP :SET LOOP ADDRESS  
3454 005616 012737 021766 000672 MOV #MSG27,ERADD :SET ERROR CODE  
3455 005624 004737 016106 JSR PC,LTGER1 :GO PRINT ERROR  
3456 005630 004737 016350 LT16X: JSR PC,ITER :GO SEE IF ITERATION  
3457 005634 000137 002430 LT16XX: JMP TSCD2 :RETURN TO SCHED  
3458
```

```

3460
3461
3462
3463 005640 032777 001000 172722 LT17: BIT #1000,@SWR ;SEE IF INHIB MAN TST
3464 005646 001005 BNE LT17A ;IF NOT: BR
3465 005650 005737 000740 TST STFLG ;SEE IF SINGLE TEST
3466 005654 001435 BEQ LT17XX ;IF NOT: BR
3467 005656 000137 016420 JMP INMT ;ELSE GO PRINT INHIB MSG
3468 005662 012737 024156 000622 LT17A: MOV #MSLT17,EMADDR ;SET TEST HEADER
3469 005670 012704 026306 MOV #MSG4,R4
3470 005674 004737 016702 JSR PC,INST ;GO PRINT INSTRUCTION
3471 005700 004737 016450 LT17IT: JSR PC,INIT2 ;SELECT DRIVE, SLAVE
3472 005704 012701 110701 MOV #110701,R1 ;SET TEST WORD
3473 005710 017702 172606 MOV @DS,R2 ;READ STATUS
3474 005714 042702 000040 BIC #40,R2 ;CLEAR PE, IF UP
3475 005720 020102 CMP R1,R2 ;SEE IF EXPT=RCVD
3476 005722 001410 BEQ LT17X ;IF SO: BR
3477 005724 012737 005700 000712 MOV #LT17IT,SCOLP ;SET LOOP ADDRESS
3478 005732 012737 021766 000672 MOV #MSG27,ERADD ;SET ERROR CODE
3479 005740 004737 016106 JSR PC,LTGER1 ;YES PRINT ERROR
3480 005744 004737 016350 LT17X: JSR PC,ITER ;GO SEE IF ITERATIONS
3481 005750 000137 002430 LT17XX: JMP TSCD2 ;RETURN TO SCHED
  
```

3483
 3484
 3485
 3486
 3487
 3488
 3489
 3490
 3491
 3492
 3493
 3494
 3495
 3496
 3497
 3498
 3499
 3500
 3501
 3502
 3503
 3504
 3505
 3506
 3507
 3508
 3509
 3510
 3511
 3512
 3513
 3514
 3515

005754 012737 024224 000622
 005762 012737 006002 000712
 005770 012700 000022
 005774 012737 000544 000674
 006002 004737 016440
 006006 012777 177777 172476
 006014 012701 000001
 006020 117777 172650 172462
 006026 017702 172472
 006032 030102
 006034 001011
 006036 012737 026763 000672
 006044 012737 000001 000716
 006052 004737 014772
 006056 000404
 006060 020102
 006062 001402
 006064 004737 014760
 006070 005300
 006072 001403
 006074 005237 000674
 006100 000740
 006102 004737 016350
 006106 004737 015440
 006112 000137 002430

```

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

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

LT20:  MOV    #MSLT20,EMADDR  :SET TEST HEADER
      MOV    #LT20A,SCOLP    :SET LOOP ADDRESS
LT20IT: MOV    #22,R0        :SET NUMBER OF ILL CODES
      MOV    #ILFT,TEMP1     :POINT TO START IF TABLE
LT20A:  JSR    PC,INIT1      :GO INIT, SELECT SLAVE + DRIVE
      MOV    #-1,@WC        :SET WC= -1
      MOV    #1,R1          :SET TEST WORD
      MOVSB  @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,LTGERO     :GO PRINT ERROR
      BR     LT20C
LT20B:  CMP    R1,R2        :SEE UNEXPECTED ERRORS
      BEQ   LT20C         :IF NOT: BR
      JSR    PC,LTGER3     :ELSE PRINT ERROR
LT20C:  DEC    R0           :SEE IF DONE ALL ILL CODES
      BEQ   LT20X         :IF SO: BR
      INC   TEMP1         :BUMP ADDRESS
      BR    LT20A         :CONTINUE
LT20X:  JSR    PC,ITER     :GO SEE IF ITERATION
      JSR    PC,DRVCLR
      JMP   TSCD2         :RETURN TO SCHED
  
```



```

3517
3518
3519
3520 006116 012737 024303 000622 LT21: MOV #MSLT21,EMADDR ;SET TEST HEADER
3521 006124 012737 006132 000712      MOV #LT21IT,SCOLP ;SET SCOPE LOOP ADDRESS
3522 006132 004737 016440      LT21IT: JSR PC,INIT1 ;GO INIT, SELECT SLAVE, DRIVE
3523 006136 052777 000300 172376      BIS #300,@TC ;SET FORMAT
3524 006144 012777 000015 172362      MOV #15,@MR ;SET WAM3
3525 006152 012777 000071 172330      MOV #71,@C1 ;SET READ+GO
3526 006160 005077 172332      CLR @FC ;ATTEMPT WRITE TO FC
3527 006164 012701 000004      MOV #4,R1 ;SET TEST WORD
3528 006170 017702 172330      MOV @ER,R2 ;GET ER
3529 006174 030102      BIT R1,R2 ;SEE IF EXPT=RCVD
3530 006176 001011      BNE LT21A ;IF SO: BR
3531 006200 012737 026777 000672      MOV #TMS19,ERADD ;SET ERROR CODE
3532 006206 012737 000001 000716      MOV #1,EXFL ;SET EXPT FLG
3533 006214 004737 014772      JSR PC,LTGERO ;GO PRINT ERROR
3534 006220 000404      BR LT21B
3535 006222 020102      LT21A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3536 006224 001402      BEQ LT21B ;IF NOT: BR
3537 006226 004737 014760      JSR PC,LTGER3 ;ELSE GO PRINT ERROR
3538 006232 004737 016350      LT21B: JSR PC,ITER ;GO SEE IF ITERATION
3539 006236 012703 040000      MOV #40000,R3
3540 006242 005303      LT21XA: DEC R3 ;DELAY FOR ALPHA
3541 006244 001376      BNE LT21XA
3542 006246 004737 014624      JSR PC,EORPA ;GO DO EOR CLEAR
3543 006252 004737 015440      JSR PC,DRVCLR
3544 006256 000137 002430      JMP TSCD2 ;RETURN TO SCHED
  
```

```

3546
3547 ;LOGIC TEST 22: CONTROL BUS PARITY (CPAR)*****
3548
3549 006262 012737 024337 000622 LT22: MOV #MSLT22,EMADDR ;SET TEST HEADER
3550 006270 012737 006276 000712 MOV #LT22IT,SCOLP ;SET SCOPE LOOP ADDRESS
3551 006276 004737 016440 LT22IT: JSR PC,INIT1 ;INIT, SELECT SLAVE+DRIVE
3552 006302 052777 000020 172210 BIS #20,@CS ;ENABLE EVEN PARITY ON MB
3553 006310 012777 177777 172200 MOV #-1,@FC ;WRITE TO FC
3554 006316 012701 000010 MOV #10,R1 ;SET TEST WORD
3555 006322 042777 000020 172170 BIC #20,@CS ;RESET PARITY TO ODD
3556 006330 017702 172170 MOV @ER,R2 ;GET ER
3557 006334 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
3558 006336 001011 BNE LT22A ;IF SO: BR
3559 006340 012737 027005 000672 MOV #TMS20,ERADD ;SET ERROR CODE
3560 006346 012737 000001 000716 MOV #1,EXFL ;SET EXPT FLG
3561 006354 004737 014772 JSR PC,LTGERO ;GO PRINT ERROR
3562 006360 000404 BR LT22X
3563 006362 020102 LT22A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3564 006364 001402 BEQ LT22X ;IF NOT: BR
3565 006366 004737 014760 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
3566 006372 004737 016350 LT22X: JSR PC,ITER ;GO SEE IF ITERATION
3567 006376 004737 015440 JSR PC,DRVCLR
3568 006402 000137 002430 JMP TSCD2 ;RETURN TO SCHED
  
```

```

3570
3571
3572
3573 006406 012737 024374 000622 LT23:  MOV #MSLT23,EMADDR ;SET TEST HEADER
3574 006414 012737 006422 000712      MOV #LT23IT,SCOLP ;SET SCOPE ADDRESS
3575 006422 004737 016440      LT23IT: JSR PC,INIT1 ;GO INIT SELECT DRIVE+SLAVE
3576 006426 042777 000360 172106      BIC #360,@TC ;SET ILLEGAL FORMAT
3577 006434 012701 000020      MOV #20,R1 ;SET TEST WORD
3578 006440 012777 000015 172066      MOV #15,@MR ;SET WAM 3
3579 006446 012777 000071 172034      MOV #71,@C1 ;SET READ+GO
3580 006454 017702 172044      MOV @ER,R2 ;READ ER
3581 006460 030102      BIT R1,R2 ;SEE IF EXPT=RCVD
3582 006462 001011      BNE LT23A ;IF SO: BR
3583 006464 012737 027014 000672      MOV #TMS21,ERADD ;SET ERROR CODE
3584 006472 012737 000001 000716      MOV #1,EXFL ;SET EXPT FLG
3585 006500 004737 014772      JSR PC,LTGERO ;GO PRINT ERROR
3586 006504 000404      BR LT23X
3587 006506 020102      LT23A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3588 006510 001402      BEQ LT23X ;IF NOT: BR
3589 006512 004737 014760      JSR PC,LTGER3 ;ELSE GO PRINT ERROR
3590 006516 004737 016350      LT23X: JSR PC,ITER ;GO SEE IF ITERATION
3591 006522 004737 014624      JSR PC,EORPA
3592 006526 004737 015440      JSR PC,DRVCLR
3593 006532 000137 002430      JMP TSCD2 ;RETURN TO SCHED
  
```

```

3595                               ;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)*****
3596
3597 006536 012737 024441 000622 LT24:  MOV    #MSLT24,EMADDR ;SET TEST HEADER
3598 006544 012737 006552 000712      MOV    #LT24IT,SCOLP ;SET SCOPE ADDRESS
3599 006552 012737 000005 000406 LT24IT: MOV    #5,ITAMT
3600 006560 004737 016466      JSR    PC,INIT3 ;GO INIT, SELECT DRIVE+SLAVE
3601 006564 052777 000300 171750      BIS    #300,@TC ;SET NORMAL FORMAT
3602 006572 012777 027314 171714      MOV    #WDATA,@BA ;SET BA
3603 006600 012777 177760 171710      MOV    #-20,@FC ;SET FC
3604 006606 012777 177770 171676      MOV    #-10,@WC ;SET WC
3605 006614 012777 000013 171712      MOV    #13,@MR ;SELECT WAM 2
3606 006622 012777 000061 171660      MOV    #61,@C1 ;SET WRITE+GO
3607 006630 052777 000020 171662      BIS    #20,@CS ;FORCE EVEN PARITY
3608 006636 012701 000040      MOV    #40,R1 ;SET TEST WORD
3609 006642 012703 000004      MOV    #4,R3
3610 006646 005000      CLR    R0
3611 006650 005300      1$:  DEC    R0
3612 006652 001376      BNE    1$ ;DELAY
3613 006654 005303      DEC    R3
3614 006656 001374      BNE    1$
3615 006660 012700 000004      MOV    #4,R0
3616 006664 012777 000013 171642 LT24B: MOV    #13,@MR ;CLOCK MR 4 TIMES
3617 006672 005300      DEC    R0
3618 006674 022700 000002      CMP    #2,R0 ;SEE IF DONE 1 BYTE
3619 006700 001002      BNE    LT24B0 ;IF NOT: BR
3620 006702 017701 171626      MOV    @MR,R1 ;ELSE GET BYTE 1
3621 006706 005700      LT24B0: TST   R0 ;SEE IF BYTE 2
3622 006710 001365      BNE    LT24B ;IF NOT: BR
3623 006712 017704 171616      MOV    @MR,R4 ;GET BYTE 2
3624 006716 005000      CLR    R0
3625 006720 005300      LT24C: DEC    R0
3626 006722 001376      BNE    LT24C ;DELAY
3627 006724 032777 000040 171572      BIT    #40,@ER ;SEE IF DPAR IS SET
3628 006732 001023      BNE    LT24D ;IF SO: BR
3629 006734 000301      SWAB  R1
3630 006736 042701 177400      BIC    #177400,R1 ;GET LOW BYTE
3631 006742 042704 000377      BIC    #377,R4
3632 006746 050401      BIS    R4,R1 ;GET HIGH BYTE
3633 006750 005237 000744      INC    T24FL ;SET T24 FLAG
3634 006754 012737 027022 000672      MOV    #TMS22,ERADD ;SET ERROR CODE
3635 006762 012737 000001 000716      MOV    #1,EXFL ;SET EXPT FLG
3636 006770 004737 014772      JSR    PC,LTGERO ;GO PRINT ERROR
3637 006774 005037 000744      CLR    T24FL ;CLEAR FLAG
3638 007000 000412      BR    LT24X
3639 007002 012701 000050      LT24D: MOV    #50,R1
3640 007006 017702 171512      MOV    @ER,R2 ;GET ERROR REGISTER
3641 007012 042702 020000      BIC    #20000,R2 ;MASK OPI
3642 007016 020102      CMP    R1,R2 ;SEE IF UNEXPECTED ERRORS
3643 007020 001402      BEQ   LT24X ;IF NOT: BR
3644 007022 004737 014760      JSR    PC,LTGER3 ;ELSE GO PRINT ERROR
3645 007026 042777 000020 171464 LT24X: BIC    #20,@CS ;RESET EVEN PARITY
3646 007034 004737 014624      JSR    PC,EORPA ;GO DO EOR CLEAR
3647 007040 004737 015440      JSR    PC,DRVCLR ;GO SEE IF DRIVE CLEAR OK
3648 007044 004737 016350      JSR    PC,ITER ;GO SEE IF ITERATION
3649 007050 012737 000020 000606      MOV    #20,ITAMT
3650 007056 000137 002430      JMP    TSCD2 ;RETURN TO SCHED

```

```

3652
3653
3654
3655 007062 012737 024501 000622 LT25: MOV #MSLT25,EMADDR ;SET TEST HEADER
3656 007070 004737 016466 LT25IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
3657 007074 052777 000300 171440 BIS #300,@TC ;SET NORMAL FORMAT
3658 007102 012777 177777 171406 MOV #-1,@FC ;SET ITLLEGAL FC
3659 007110 012777 000013 171416 MOV #13,@MR ;SET WAM 2
3660 007116 012777 000061 171364 MOV #61,@C1 ;LOAD WRITE+GO
3661 007124 012701 004000 MOV #4000,R1 ;SET TEST WORD
3662 007130 017702 171370 MOV @ER,R2 ;GET ER
3663 007134 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
3664 007136 001014 BNE LT25A ;IF SO: BR
3665 007140 012737 007070 000712 MOV #LT25IT,SCOLP ;SET LOOP ADDRESS
3666 007146 012737 027110 000672 MOV #TMS31,ERADD ;SET ERROR CODE
3667 007154 012737 000001 000716 MOV #1,EXFL ;SET EXPT FLAG
3668 007162 004737 014772 JSR PC,LTGERO ;GO PRINT ERROR
3669 007166 000404 BR LT25X
3670 007170 020102 LT25A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3671 007172 001402 BEQ LT25X ;IF NOT: BR
3672 007174 004737 014760 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
3673 007200 004737 016350 LT25X: JSR PC,ITER ;GO SEE IF ITERATION
3674 007204 004737 015440 JSR PC,DRVCLR
3675 007210 000137 002430 JMP TSCD2 ;RETURN TO SCHED
  
```

```

3677
3678
3679
3680 007214 012737 024535 000622 LT26: MOV #MSLT26,EMADDR ;SET TEST HEADER
3681 007222 004737 016466 LT26IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
3682 007226 005000 CLR R0
3683 007230 005300 1$: DEC R0
3684 007232 001376 BNE 1$ ;AWAIT OPI RESET
3685 007234 052777 000300 171300 BIS #300,@TC ;SET NORMAL FORMAT
3686 007242 012777 177770 171242 MOV #-10,@WC ;SET WC=-10
3687 007250 012777 177760 171240 MOV #-20,@FC ;SET FC=-20
3688 007256 012777 000013 171250 MOV #13,@MR ;SET WAM 3
3689 007264 012777 000061 171216 MOV #61,@C1 ;LOAD WRITE+GO
3690 007272 012701 001000 MOV #1000,R1 ;SET TEST WORD
3691 007276 005000 CLR R0
3692 007300 005300 2$: DEC R0
3693 007302 001376 BNE 2$ ;DELAY
3694 007304 012777 000025 171222 MOV #25,@MR ;LOAD MM EOR CLEAR
3695 007312 105077 171216 CLR# @MR ;RESET MR
3696 007316 012703 000004 MOV #4,R3
3697 007322 005000 CLR R0
3698 007324 032777 001000 171172 3$: BIT #1000,@ER ;SEE IF FCE SET
3699 007332 001022 BNE 4$ ;IF SO: BR
3700 007334 005300 DEC R0
3701 007336 001372 BNE 3$ ;DELAY
3702 007340 005303 DEC R3
3703 007342 001370 BNE 3$
3704 007344 017702 171154 MOV @ER,R2 ;GET ER
3705 007350 012737 007222 000712 MOV #LT26IT,SCOLP ;SET SCOPE ADDRESS
3706 007356 012737 027067 000672 MOV #TMS28,ERADD
3707 007364 012737 000001 000716 MOV #1,EXFL ;SET EXPT FLG
3708 007372 004737 014772 JSR PC,LTGER0 ;GO PRINT ERROR
3709 007376 000406 BR LT26X
3710 007400 017702 171120 4$: MOV @ER,R2 ;GET ERROR REGISTER
3711 007404 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3712 007406 001402 BEQ LT26X ;IF NOT: BR
3713 007410 004737 014760 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
3714 007414 004737 016350 LT26X: JSR PC,ITER ;GO SEE IF ITERATION
3715 007420 004737 015440 JSR PC,DRVCLR
3716 007424 000137 002430 JMP TSCD2 ;RETURN TO SCHED

```

```

3718
3719
3720 ;LOGIC TEST 27: ILLEGAL REGISTER(ILR)*****
3721 007430 022737 172400 000510 LT27: CMP #172400,C1 ;SEE IF ADDRESSES OPEN
3722 007436 000442 BR LT27XX ;THIS TEST IS BYPASSED DUE TO THE REQUIREMENT
3723 ;OF INSERTING AND REMOVEING ADDRESS JUMPERS
3724 ;IN THE RH01/RH70 TO OPERATE.
3725 007440 001041 BNE LT27XX ;IF NOT: BR
3726 007442 012737 007466 000712 MOV #LT27A,SCOLP ;SET SCOPE ADDRESS
3727 007450 012737 024571 000622 MOV #MSLT27,EMADDR ;SET TEST HEADER
3728 007456 012700 000020 LT27IT: MOV #20,R0 ;SET NUMBER OF ILR TESTS
3729 007462 012701 172434 MOV #172434,R1 ;SET FIRST ILR ADDRESS
3730 007466 004737 016466 LT27A: JSR PC,INIT3 ;GO INIT, SELECT DRIVE+SLAVE
3731 007472 011103 MOV (R1),R3 ;ATTEMPT ILR READ
3732 007474 032777 000002 171022 BIT #2,@ER ;SEE IF ILR=1
3733 007502 001010 BNE LT27B ;IF SO: BR
3734 007504 012737 000001 000716 MOV #1,EXFL ;SET EXPT-NOT RCVD FLAG
3735 007512 012737 026711 000672 MOV #TMS10,ERADD ;SET ERROR CODE
3736 007520 004737 015000 JSR PC,LTGER ;GO PRINT ERROR
3737 007524 005300 LT27B: DEC R0 ;SEE IF DONE ALL
3738 007526 001402 BEQ LT27X ;IF SO: BR
3739 007530 005721 TST (R1)+ ;BUMP ADDRESS
3740 007532 000755 BR LT27A ;CONTINUE TESTS
3741 007534 004737 016350 LT27X: JSR PC,ITER ;GO SEE IF ITERATIONS
3742 007540 004737 015440 JSR PC,DRVCLR
3743 007544 000137 002430 LT27XX: JMP TSCD2 ;RETURN TO SCHED
  
```

```

3745
3746
3747
3748 007550 012737 027116 000672 LT30: MOV #TMS32,ERADD ;SET ERROR CODE
3749 007556 012737 024625 000622 MOV #MSLT30,EMADDR ;SET TEST HEADER
3750 007564 012737 007572 000712 MOV #LT30IT,SCOLP ;SET SCOPE ADDRESS
3751 007572 004737 016466 LT30IT: JSR PC,INIT3 ;INIT, SELECT DRIVE + SLAVE
3752 007576 052777 000300 170736 BIS #300,@TC ;SET NORMAL FORMAT
3753 007604 012701 010000 MOV #10000,R1 ;SET TEST WORD
3754 007610 012777 000017 170716 MOV #17,@MR ;CRIPPLE OCCUPIED
3755 007616 005077 170674 CLR @FC ;SET FC3
3756 007622 012777 000061 170660 MOV #61,@C1 ;LOAD WRITE+GO
3757 007630 032777 010000 170666 BIT #10000,@ER ;SEE IF DTE SET
3758 007636 001005 BNE LT30A ;IF SO: BR
3759 007640 012737 000001 000716 MOV #1,EXFL ;SET EXPT FLG
3760 007646 004737 014772 JSR PC,LTGERO ;GO PRINT ERROR
3761 007652 004737 016466 LT30A: JSR PC,INIT3 ;GO INIT SELECT DRIVE,SLAVE
3762 007656 052777 000300 170656 BIS #300,@TC ;SET FORMAT
3763 007664 012701 010000 MOV #10000,R1 ;SET TEST WORD
3764 007670 005077 170622 CLR @FC ;SET FCS
3765 007674 012777 000015 170632 MOV #15,@MR ;SET WRAP 3
3766 007702 012777 000061 170600 MOV #61,@C1 ;LOAD WRITE+GO
3767 007710 012704 040000 MOV #40000,R4
3768 007714 005777 170622 LT30B: TST @TC ;SEE IF ALPHA
3769 007720 100015 BPL LT30C ;AWAIT ALPHA
3770 007722 005300 DEC R0
3771 007724 001373 BNE LT30B
3772 007726 013704 000622 MOV EMADDR,R4
3773 007732 004737 017446 JSR PC,TTOUT ;PRINT HEADER
3774 007736 012704 022522 MOV #MSG50,R4
3775 007742 004737 017446 JSR PC,TTOUT ;PRINT ALPHA ERROR
3776 007746 004737 016320 JSR PC,SCOPE
3777 007752 000435 BR LT30X
3778 007754 012777 000015 170552 LT30C: MOV #15,@MR ;CLOCK MR
3779 007762 012777 000015 170544 MOV #15,@MR ;CLOCK MR
3780 007770 005000 CLR R0
3781 007772 005300 LT30D: DEC R0
3782 007774 001376 BNE LT30D ;DELAY
3783 007776 032777 010000 170520 BIT #10000,@ER ;SEE IF DTE SET
3784 010004 001006 BNE LT30E ;IF SO: BR
3785 010006 012737 000001 000716 MOV #1,EXFL ;SET EXPT FLG
3786 010014 004737 014772 JSR PC,LTGERO ;GO PRINT ERROR
3787 010020 000412 BR LT30X
3788 010022 012701 010000 LT30E: MOV #10000,R1 ;SET TEST WORD
3789 010026 017702 170472 MOV @ER,R2 ;GET ERROR REGISTER
3790 010032 042702 020100 BIC #20100,R2 ;MASK OPI AND VPE
3791 010036 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3792 010040 001402 BEQ LT30X ;IF NOT: BR
3793 010042 004737 014760 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
3794 010046 004737 016350 LT30X: JSR PC,ITER ;GO SEE IF ITERATION
3795 010052 004737 014624 JSR PC,EORPA ;GO CLEAR GO BIT
3796 010056 004737 015440 JSR PC,DRVCLR
3797 010062 000137 002430 JMP TSCD2 ;RETURN TO SCHED
3798

```


3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855

010066 012737 024663 000622
010074 012737 010074 000712
010102 012737 027132 000672
010110 012737 000002 000606
010116 004737 016466
010122 005000
010124 005300
010126 001376
010130 052777 000300 170404
010136 012777 000013 170370
010144 005077 170346
010150 012705 020000
010154 012702 010172
010160 004737 010372
010164 012777 000061 170316
010172 030577 170326
010176 001002
010200 000163 010464
010204 017702 170314
010210 020502
010212 001403
010214 004737 014760
010220 000453
010222 004737 010556
010226 102450
010230 012737 010244 000712
010236 012737 027146 000672
010244 004737 016466
010250 005000
010252 005300
010254 001376
010256 052777 000300 170256
010264 012777 000015 170242
010272 012702 010314
010276 012705 020000
010302 004737 010372
010306 012777 000071 170174
010314 030577 170204
010320 001002
010322 000163 010464
010326 017702 170172
010332 020502
010334 001403
010336 004737 014760
010342 000402
010344 004737 010556
010350 004737 016350
010354 004737 015440
010360 012737 000020 000606
010366 000137 002430

```

;LOGIC TEST 31: OPERATION INCOMPLETE(OPI)*****
LT31:  MOV    #MSLT31,EMADDR ;SET TEST HEADER
LT31IT: MOV    #LT31IT,SCOLP ;SET SCOPE ADDRESS
        MOV    #TMS33A,ERADD ;SET ERROR MSG HDR
        MOV    #2,ITAMT ;SET REDUCED ITER COUNT
        JSR   PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
        CLR   RO
1$:    DEC   RO
        BNE  1$ ;AWAIT OPI RESET
        BIS  #300,@TC ;SET FORMAT
        MOV  #13,@MR ;SET WAM 2
        CLR  @FC ;SET FRAME COUNT
        MOV  #20000,R5 ;SET TEST BIT (OPI)
        MOV  #2$,R2 ;SET RETURN ADDRESS FROM TIMER
        JSR  PC,TIMON ;START TIMER
        MOV  #61,@C1 ;LOAD WRITE+GO
2$:    BIT   R5,@ER ;BRANCH WHEN OPI SETS
        BNE  3$
        JMP  TIMER(R3) ;GO TO TIMER & RETURN TO 2$ ABOVE
3$:    MOV   @ER,R2 ;GET ERROR REGISTER
        CMP  R5,R2 ;SEE IF UNEXPECTED ERRORS
        BEQ  4$ ;IF NOT: BR
        JSR  PC,LTGER3 ;ELSE PRINT ERROR
        BR   LT31X
4$:    JSR  PC,TIMOK ;GO CHECK TIME FOR OPI TO SET
        BVS  LT31X ;BRANCH IF TIME WAS INCORRECT

        MOV  #LT31A,SCOLP ;SET SCOPE LOOP
        MOV  #TMS33B,ERADD ;SET ERROR MSG HEADER
LT31A: JSR   PC,INIT3 ;GO INIT
        CLR   RO
1$:    DEC   RO ;WAIT FOR OPI TO CLEAR
        BNE  1$
        BIS  #300,@TC ;SET FORMAT
        MOV  #15,@MR ;SET WRAP 3
        MOV  #2$,R2 ;SET RETURN ADDRESS FROM TIMER
        MOV  #20000,R5 ;SET TEST WORD
        JSR  PC,TIMON ;START TIMER
        MOV  #71,@C1 ;LOAD READ+GO
2$:    BIT   R5,@ER ;BRANCH WHEN OPI SETS
        BNE  3$
        JMP  TIMER(R3) ;GO TO TIMER
3$:    MOV   @ER,R2 ;GET ERROR REGISTER
        CMP  R5,R2 ;SEE IF UNEXPECTED ERRORS
        BEQ  4$ ;ELSE PRINT ERROR
        JSR  PC,LTGER3 ;ELSE PRINT ERROR
        BR   LT31X ;EXIT TEST
4$:    JSR  PC,TIMOK ;GO CHECK TIME
LT31X: JSR  PC,ITER ;GO SEE IF ITERATIONS
        JSR  PC,DRVCLR
        MOV  #20,ITAMT
        JMP  TSCD2 ;RETURN TO SCHED
    
```

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

```

3856 ;ENANCE REGISTER (BIT 6) THAT TOGGLES EVERY 56 (10) MICROSECONDS. THIS
3857 ;ROUTINE WAITS FOR THE OSCILLATOR TO TOGGLE AND RETURN WITH R3 INDICATING
3858 ;THE STATE OF THE OSCILLATOR.
3859 010372 005000 TIMON: CLR R0 ;CLEAR TICK COUNT
3860 010374 005001 CLR R1
3861 010376 012703 000024 MOV #24,R3 ;PRFSET INDEX TO TIMER
3862 010402 032777 000100 170124 BIT #100,AMR ;BRANCH IF OSC CLEAR
3863 010410 001405 BEQ 2$
3864 010412 032777 000100 170114 1$: BIT #100,AMR ;WAIT FOR OSC TO CLEAR
3865 010420 001374 BNE 1$
3866 010422 C0405 BR 4$ ;EXIT
3867
3868 010 OC 03 2$: NEG R3 ;SET INDEX TO TIMER
3869 010426 032777 000100 170100 3$: BIT #100,AMR ;WAIT FOR OSC TO SET
3870 010434 001774 BEQ 3$
3871 010436 000207 4$: RTS PC ;RETURN
3872
3873 ;THIS ROUTINE TIMES AN EVENT. EACH TIME THE OSCILLATOR BIT CHANGES
3874 ;STATE THE TICK COUNT IN R1 & R0 IS INCREMENTED. THE ROUTINE IS CALLED
3875 ;USING R3 AS AN INDEX TO INDICATE THE OSCILLATORS PAST STATE. WHEN
3876 ;THE OSC BIT CHANGES STATE R3 IS NEGATED.
3877 010440 032777 000100 170066 TIMER1: BIT #100,AMR ;BRANCH IF OSC HAS CHANGED STATE
3878 010446 001406 BEQ TIMER
3879 010450 000112 JMP (R2) ;RETURN
3880 010464 001406 .=TIMER1+24
3881 010464 005403 TIMER: NEG R3 ;SET INDEX TO OTHER STATE
3882 010466 062700 000001 ADD #1,R0 ;INCREMENT TICK COUNT
3883 010472 005501 ADC R1
3884 010474 022701 000003 CMP #3,R1 ;BRANCH IF TIMER OVERFLOWS
3885 010500 001410 BEQ TIMOVF
3886 010502 000112 JMP (R2) ;RETURN
3887 010510 001410 .=TIMER +24
3888 010510 032777 000100 170016 TIMERO: BIT #100,AMR ;BRANCH IF OSC SET
3889 010516 001362 BNE TIMER
3890 010520 000112 JMP (R2) ;RETURN
3891
3892 010522 013704 000622 TIMOVF: MOV EMADDR,R4 ;TYPE TEST HEADER
3893 010526 004737 017446 JSR PC,TTOUT
3894 010532 013704 000672 MOV ERADD,R4 ;GET ERROR MSG ADDRESS
3895 010536 004737 017446 JSR PC,TTOUT ;AND TYPE IT
3896 010542 012704 027226 MOV #TMS33E,R4 ;TYPE
3897 010546 004737 017446 JSR PC,TTOUT ;'TIMER OVERFLOWED'
3898 010552 000137 010350 JMP LT31X ;GO EXIT TEST
3899
3900 ;ROUTINE TO CHECK IF TIME IS WITHIN LIMITS. IF NOT THE ROUTINE RETURNS
3901 ;WITH THE 'V' BIT SET. THE LIMITS WERE SLECTED BY DIVIDING THE TIME
3902 ;IN MICROSECONDS BY 448. THE LOWER LIMIT IS 5,500,000 USECS (5.5 SECS);
3903 ;THE UPPER LIMIT IS 8,500,000 USECS (8.5 SECS). THE 448 IS DERIVED FROM
3904 ;56 USECS/TICK TIMES THE DIVISION BY 8 BY THE TIMOK ROUTINE.
3905 010556 000240 TIMOK: NOP
3906 010560 006201 ASR R1 ;DIVIDE COUNT BY 8
3907 010562 006000 ROR R0
3908 010564 006201 ASR R1
3909 010566 006000 ROR R0
3910 010570 006201 ASR R1
3911 010572 006000 ROR R0
  
```

3912	010574	020027	027764		CMP	RO,#12276.		;BRANCH IF GREATER THAN LOWER LIMIT
3913	010600	101016			BHI	1\$		
3914	010602	013704	000622		MOV	EMADDR,R4		;GET ERROR MSG HEADER
3915	010606	004737	017446		JSR	PC,TTOUT		;TYPE ERROR MSG HEADER
3916	010612	013704	000672		MOV	ERADD,R4		;GET ERROR DESCRIPTOR MSG
3917	010616	004737	017446		JSR	PC,TTOUT		
3918	010622	012704	027161		MOV	#TMS33C,R4		;TYPE 'OCCURED TOO SOON' :
3919	010626	004737	017446		JSR	PC,TTOUT		
3920	010632	000262			SEV			;SET 'V' TO INDICATE ERROR
3921	010634	000420			BR	2\$		
3922								
3923	010636	020027	045035	1\$:	CMP	RO,#18973.		;BRANCH IF LESS THAN UPPER LIMIT
3924	010642	003415			BLE	2\$		
3925	010644	013704	000622		MOV	EMADDR,R4		;GET ERROR MSG HEADER
3926	010650	004737	017446		JSR	PC,TTOUT		
3927	010654	013704	000672		MOV	ERADD,R4		
3928	010660	004737	017446		JSR	PC,TTOUT		;TYPE ERROR MSG HEADER
3929	010664	012704	027203		MOV	#TMS33D,R4		;TYPE 'OCCURED TOO LATE'
3930	010670	004737	017446		JSR	PC,TTOUT		
3931	010674	000262			SEV			
3932	010676	000207		2\$:	RTS	PC		

```

3934
3935
3936
3937
3938
3939 010700 012737 024717 000622 LT32:  MOV    #MSLT32,EMADDR ;SET TEST HEADER
3940 010706 012737 010714 000712      MOV    #LT32IT,SCOLP ;SET SCOPE ADDRESS
3941 010714 004737 016466      LT32IT: JSR    PC,INIT3 ;INIT, SELECT DRIVE +SLAVE
3942 010720 013700 000664      MOV    SLVN,RO ;GET SLAVE NUMBER
3943 010724 005100      COM    RO ;SET NONEXISTANT SLAVE
3944 010726 042700 177770      BIC    #177770,RO ;MASK SLAVE NUMBER
3945 010732 052700 000300      BIS    #300,RO ;SET FORMAT
3946 010736 010077 167600      MOV    RO,@TC ;SELECT ILLEGAL SLAVE
3947 010742 032777 002000 167566      BIT    #2000,@DT ;EXIT TEST IF SALVE AVAILABLE
3948 010750 001032      BNE    LT32XX
3949 010752 012777 000071 167530      MOV    #71,@C1 ;LOAD READ+GO
3950 010760 012701 044000      MOV    #44000,R1 ;SET TEST WORD
3951 010764 017702 167534      MOV    @ER,R2 ;READ ER
3952 010770 030102      BIT    R1,R2 ;SEE IF EXPT=RCVD
3953 010772 001011      BNE    1$ ;IF SO: BR
3954 010774 012737 027246 000672      MOV    #TMS34,ERADD ;SET ERROR CODE
3955 011002 012737 000001 000716      MOV    #1,EXFL ;SET ERROR CODE
3956 011010 004737 014772      JSR    PC,LTGERO ;GO PRINT ERROR
3957 011014 000404      BR    LT32X
3958 011016 020102      1$:  CMP    R1,R2 ;SEE IF UNEXPECTED ERRORS
3959 011020 001402      BEQ    LT32X ;IF NOT: BR
3960 011022 004737 014760      JSR    PC,LTGER3 ;ELSE PRINT ERROR
3961 011026 004737 016350      LT32X: JSR    PC,ITER ;GO SEE IF ITERATIONS
3962 011032 004737 015440      JSR    PC,DRVCLR
3963 011036 000137 002430      LT32XX: JMP   TSCD2 ;RETURN TO SCHED
  
```

3965
 3966
 3967
 3968
 3969
 3970
 3971
 3972
 3973
 3974
 3975
 3976
 3977
 3978
 3979
 3980
 3981
 3982
 3983
 3984
 3985

011042	012737	024753	000622
011050	012737	011056	000712
011056	004737	016466	
011062	012777	000013	167444
011070	012777	177777	167420
011076	012777	000031	167404
011104	032777	020000	167410
011112	001010		
011114	012737	026741	000672
011122	012737	000001	000716
011130	004737	014772	
011134	004737	016350	
011140	C00137	002430	

: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,INIT3	: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

3987
 3988
 3989
 3990
 3991
 3992
 3993
 3994
 3995
 3996
 3997
 3998
 3999
 4000
 4001
 4002
 4003
 4004
 4005
 4006
 4007
 4008

011144 012737 026661 000672
 011152 012737 025007 000622
 011160 012700 000004
 011164 004737 016466
 011170 042777 003400 167344
 011176 052777 001400 167336
 011204 032777 000040 167310
 011212 001410
 011214 012737 000002 000716
 011222 012737 011164 000712
 011230 004737 014772
 011234 004737 016502
 011240 032777 000040 167254
 011246 001010
 011250 012737 011240 000712
 011256 012737 000001 000716
 011264 004737 014772
 011270 004737 016350
 011274 000137 002430

;LOGIC TEST 34: PHASE ENCODED STATUS(PES)*****

LT34: MOV #TMS6,ERADD ;SET ERROR CODE
 MOV #MSLT34,EMADDR ;SET TEST HEADER
 LT34IT: MOV #4,R0
 LT34A1: JSR PC,INIT3 ;GO INIT, SELECT DRIVE+SLAVE
 BIC #3400,@TC ;SELECT NRZI
 BIS #1400,@TC
 LT34A: BIT #40,@DS ;SEE IF PES=0
 BEQ LT34B ;IF SO: BR
 MOV #2,EXFL ;SET RCVD-NOT EXPT
 MOV #LT34A1,SCOLP ;SET SCOPE ADDRESS
 JSR PC,LTGERO ;GO PRINT ERROR
 LT34B: JSR PC,INIT4
 LT34C: BIT #40,@DS ;SEE IF PES=1
 BNE LT34X ;IF SO: BR
 MOV #LT34C,SCOLP ;SET SCOPE ADDRESS
 MOV #1,EXFL ;SET EXPT-NOT RCVD FLAG
 JSR PC,LTGERO ;GO PRINT ERROR
 LT34X: JSR PC,ITER ;GO SEE IF ITERATION
 LT34XX: JMP TSCD2 ;RETURN TO SCHED

```

4010
4011
4012
4013 011300 012737 027271 000672 LT35:  MOV      #TMS37,ERADD
4014 011306 012737 025043 000622      MOV      #MSLT35,EMADDR
4015 011314 004737 016466          LT35IT: JSR      PC,INIT3      ;INIT SELECT DRIVE, SLAVE
4016 011320 032777 000020 167174 1$:      BIT      #20,@DS      ;SEE IF SDWN IS RESET
4017 011326 001374          BNE      1$      ;IF NOT: BR
4018 011330 052777 000300 167204      BIS      #300,@TC      ;SET FORMAT
4019 011336 012777 000015 167170      MOV      #15,@MR      ;SET WAM 3
4020 011344 012777 000071 167136      MOV      #71,@C1      ;LOAD READ+GO
4021 011352 032777 020000 167162      BIT      #20000,@TC    ;SEE IF SAC=0
4022 011360 001410          BEQ      LT35A      ;IF SO: BR
4023 011362 012737 000002 000716      MOV      #2,EXFL      ;SET RCV-NOT EXPT FLAG
4024 011370 012737 011314 000712      MOV      #LT35IT,SCOLP ;SET SCOPE ADDRESS
4025 011376 004737 014772          JSR      PC,LTGERO    ;GO PRINT ERROR
4026 011402 004737 016466          LT35A:  JSR      PC,INIT3    ;INIT
4027 011406 005277 167130          INC      @TC          ;BUMP SLAVE ADDRESS
4028 011412 032777 020000 167122      BIT      #20000,@TC    ;SEE IF SAC=1
4029 011420 001010          BNE      LT35X      ;IF SO: BR
4030 011422 012737 011402 000712      MOV      #LT35A,SCOLP ;SET SCOPE ADDRESS
4031 011430 012737 000001 000716      MOV      #1,EXFL      ;SE EXPT-NOT RCVD FLAG
4032 011436 004737 014772          JSR      PC,LTGERO    ;GO PRINT ERROR
4033 011442 004737 016350          LT35X:  JSR      PC,ITER    ;
4034 011446 000137 002430          JMP      TSCD2      ;RETURN TO SCHED
  
```

```

4036
4037
4038
4039 011452 012737 025110 000622 LT36:  MOV #M$LT36,EMADDR
4040 011460 012737 027277 000672      MOV #TMS38,ERADD      ;SET ERROR CODE
4041 011466 004737 016466      LT36IT: JSR PC,INIT3   ;INIT, SELECT DRIVE+SLAVE
4042 011472 032777 040000 167042      BIT #40000,@TC      ;SEE IF FCS=0
4043 011500 001410      BEQ 1$              ;IF SO: BR
4044 011502 012737 011466 000712      MOV #LT36IT,SCOLP   ;SET SCOPE ADDRESS
4045 011510 012737 000002 000716      MOV #2,EXFL         ;SET RCVD-NOT EXPT
4046 011516 004737 014772      JSR PC,LTGERO      ;GO PRINT ERROR
4047 011522 004737 016466      1$: JSR PC,INIT3     ;INIT
4048 011526 005077 166764      CLR @FC            ;WRITE TO FC
4049 011532 032777 040000 167002      BIT #40000,@TC     ;SEE IF FCS=1
4050 011540 001010      BNE LT36X          ;IF SO: BR
4051 011542 012737 011522 000712      MOV #1$,SCOLP      ;SET SCOPE ADDRESS
4052 011550 012737 000001 000716      MOV #1,EXFL        ;SET EXPT-NOT RCVD
4053 011556 004737 014772      JSR PC,LTGERO      ;GO PRINT ERROR
4054 011562 004737 016350      LT36X: JSR PC,ITER   ;
4055 011566 000137 002430      JMP TSCD2          ;RETURN TO SCHED
  
```



```

4057
4058
4059
4060 011572 012737 025155 000622 LT37: MOV #MSLT37,EMADDR
4061 011600 012737 027305 000672 MOV #TMS39,ERADD ;SET ERROR CODE
4062 011606 004737 016466 LT37IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
4063 011612 052777 000300 166722 BIS #300,@TC ;SET FORMAT
4064 011620 005777 166716 TST @TC ;SEE IF ACCL=1
4065 011624 100410 BMI LT37A ;IF SO: BR
4066 011626 012737 000001 000716 MOV #1,EXFL
4067 011634 012737 011606 000712 MOV #LT37IT,SCOLP ;SET SCOPE ADDRESS
4068 011642 004737 014772 JSR PC,LTGERO ;GO PRINT ERROR
4069 011646 004737 016466 LT37A: JSR PC,INIT3 ;INIT
4070 011652 052777 000300 166662 BIS #300,@TC ;SET FORMAT
4071 011660 012777 000015 166646 MOV #15,@MR ;SET WAM 3
4072 011666 012777 000071 166614 MOV #71,@C1 ;LOAD READ+GO
4073 011674 012700 100000 MOV #100000,RO ;SET ACCL DELAY
4074 011700 005777 166636 LT37B: TST @TC ;SEE IF ACCL=0
4075 011704 100012 BPL LT37X ;IF SO: BR
4076 011706 005300 DEC RO
4077 011710 001373 BNE LT37B ;DELAY
4078 011712 012737 011646 000712 MOV #LT37A,SCOLP ;SET SCOPE ADDRESS
4079 011720 012737 000002 000716 MOV #2,EXFL
4080 011726 004737 014772 JSR PC,LTGERO ;GO PRINT ERROR
4081 011732 004737 016350 LT37X: JSR PC,ITER
4082 011736 000137 002430 JMP TSCD2 ;RETURN TO SCHED
  
```

```

4084
4085
4086
4087 011742 012737 011756 000712 LT40: MOV #LT40IT,SCOLP ;SET SCOPE ADDRESS
4088 011750 012737 025223 000622 MOV #MSLT40,EMADDR
4089 011756 004737 016502 LT40IT: JSR PC,INIT4 ;INIT, SELECT DRIVE+SLAVE
4090 011762 005000 CLR R0
4091 011764 005300 1$: DEC R0
4092 011766 001376 BNE 1$ ;DELAY FOR OPI RESET
4093 011770 052777 002300 166544 BIS #2300,@TC
4094 011776 012777 000007 166530 MOV #7,@MR ;SET WAM 0
4095 012004 012777 000027 166476 MOV #27,@C1 ;LOAD WRITE TAPE MARK+GO
4096 012012 012700 100000 MOV #100000,R0 ;SET DELAY
4097 012016 032777 000004 166476 2$: BIT #4,@DS ;SEE IF TM=1
4098 012024 001012 BNE LT40X ;IF SO: BR
4099 012026 005300 DEC R0
4100 012030 001372 BNE 2$ ;DELAY
4101 012032 012737 026637 000672 MOV #TMS3,ERADD
4102 012040 012737 000001 000716 MOV #1,EXFL
4103 012046 004737 014772 JSR PC,LTGERO ;GO PRINT ERROR
4104 012052 004737 016350 LT40X: JSR PC,ITER
4105 012056 000137 002430 LT40XX: JMP TSCD2 ;RETURN TO SCHED
  
```

```

4107
4108
4109
4110 012062 012737 012076 000712 LT41: MOV #LT41IT,SCOLP ;SET SCOPE ADDRESS
4111 012070 012737 025270 000622 MOV #MSLT41,EMADDR
4112 012076 004737 016466 LT41IT: JSR PC,INIT3 ;INIT, SELECT DRIVE,SLAVE
4113 012102 052777 001700 166432 BIS #1700,@TC ;SET NRZ+NORMAL FORMAT
4114 012110 012777 177760 166400 MOV #-20,@FC ;SET FCS
4115 012116 012777 000007 166410 MOV #7,@MR ;SET WAM 0
4116 012124 012777 000027 166356 MOV #27,@C1 ;LOAD WRITE TAPE MARK+GO
4117 012132 005000 CLR R0
4118 012134 032777 000004 166360 1$: BIT #4,@DS ;SEE IF TM=1
4119 012142 001012 BNE 2$ ;IF SO: BR
4120 012144 005300 DEC R0
4121 012146 001372 BNE 1$ ;DELAY
4122 012150 012737 026637 000672 MOV #TMS3,ERADD ;SET ERROR CODE
4123 012156 012737 000001 000716 MOV #1,EXFL
4124 012164 004737 014772 JSR PC,LTGERO ;GO PRINT ERROR
4125 012170 032777 002000 166326 2$: BIT #2000,@ER ;SEE IF ITM=1
4126 012176 001010 BNE 3$ ;IF SO: BR
4127 012200 012737 027102 000672 MOV #TMS30,ERADD ;SET ERROR CODE
4128 012206 012737 000001 000716 MOV #1,EXFL
4129 012214 004737 014772 JSR PC,LTGERO ;GO PRINT ERROR
4130 012220 032777 000100 166276 3$: BIT #100,@ER ;SEE IF VPE=1
4131 012226 001011 BNE 4$ ;IF SO: BR
4132 012230 012737 027067 000672 MOV #TMS28,ERADD ;SET ERROR CODE
4133 012236 012737 000001 000716 MOV #1,EXFL
4134 012244 004737 014772 JSR PC,LTGERO ;GO PRINT ERROR
4135 012250 000410 BR LT41X
4136 012252 012701 002100 4$: MOV #2100,R1 ;SET EXPT ERROR BITS
4137 012256 017702 166242 MOV @ER,R2 ;GET ERROR REGISTER
4138 012262 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
4139 012264 001402 BEQ LT41X ;IF NOT: BR
4140 012266 004737 014760 JSR PC,LTGER3 ;ELSE PRINT ERROR
4141 012272 005002 LT41X: CLR R2 ;SET TIMER
4142 012274 032777 000200 166220 1$: BIT #200,@DS ;SEE IF DRY SET
4143 012302 001002 BNE 2$ ;IF SO: BR
4144 012304 005302 DEC R2 ;AWAIT DRY
4145 012306 001372 BNE 1$ ;DELAY
4146 012310 004737 016350 2$: JSR PC,ITER ;GO SEE IF ITERATIONS
4147 012314 004737 015440 JSR PC,DRVCLR ;GO DO DRIVE CLEAR
4148 012320 000137 002430 JMP TSCD2 ;RETURN TO SCHED

```

```

4150
4151
4152
4153
4154
4155
4156
4157 012324 012737 001700 000776 LT42: MOV #1700, UDES ;SET UNIT DESCRIPTION = NRZ
4158 012332 004737 014574 JSR PC, STATIC ;GO SEE IF STATIC ONLY
4159 012336 012700 001000 MOV #1000, R0
4160 012342 005300 1$: DEC R0
4161 012344 001376 BNE 1$ ;PAUSE
4162 012346 012737 025337 000622 MOV #MSLT42, EMADDR
4163 012354 012737 01362 000712 MOV #LT42IT, SCOLP ;SET SCOPE ADDRESS
4164 012362 004737 016516 LT42IT: JSR PC, INIT ;INIT SELECT DRIVE+SLAVE
4165 012366 012777 177770 166116 MOV #-10, @WC
4166 012374 012777 177760 166114 MOV #-20, @FC ;SET FC=20
4167 012402 012777 027314 166104 MOV #WDATA, @BA ;SET BUS ADDRESS
4168 012410 012777 000007 166116 MOV #7, @MR ;SET MM CODE
4169 012416 012777 000061 166064 MOV #61, @C1 ;LOAD WRITE+GO
4170 012424 005000 CLR R0
4171 012426 032777 000200 166066 LT42A: BIT #200, @DS ;SEE IF DRY=1
4172 012434 001002 BNE LT42B ;IF SO: BR
4173 012436 005300 DEC R0
4174 012440 001372 BNE LT42A ;DELAY
4175 012442 022777 000200 166054 LT42B: CMP #200, @ER ;SEE IF LRC ERROR ONLY
4176 012450 001007 BNE LT42B1 ;IF NOT: BR
4177 012452 017702 166052 MOV @CC, R2 ;GET CHECK CHAR
4178 012456 042702 177000 BIC #177000, R2 ;MASK CRC
4179 012462 022702 000777 CMP #777, R2 ;SEE IF SETUP CRC IS CORRECT
4180 012466 001410 BEQ LT42B2 ;IF SO: BR
4181 012470 004737 014760 LT42B1: JSR PC, LTGER3 ;ELSE PRINT ERROR SETUP
4182 012474 012704 022613 MOV #MSG55, R4
4183 012500 004737 017446 JSR PC, TTOUT ;PRINT SETUP ERROR MSG
4184 012504 000137 002430 JMP TSCD2 ;RETURN TO SCHED
4185 012510 004737 016516 LT42B2: JSR PC, INIT ;GO INIT
4186 012514 012777 177770 165770 MOV #-10, @WC ;SET WC
4187 012522 012777 177760 165766 MOV #-20, @FC ;SET FC
4188 012530 012777 027314 165756 MOV #WDATA, @BA ;SET BA
4189 012536 012777 000021 165770 MOV #21, @MR ;SET MM
4190 012544 012777 000061 165736 MOV #61, @C1 ;LOAD WRITE+GO
4191 012552 005000 CLR R0
4192 012554 032777 000200 165740 LT42C: BIT #200, @DS ;SEE IF DRY
4193 012562 001002 BNE LT42D ;IF SO: BR
4194 012564 005300 DEC R0
4195 012566 001372 BNE LT42C ;AWAIT DRY
4196 012570 005777 165730 LT42D: TST @ER ;SEE IF CRC=1
4197 012574 100411 BMI LT42E ;IF SO: BR
4198 012576 012737 027263 000672 MOV #TMS36, ERADD ;SET ERROR CODE
4199 012604 012737 000001 000716 MOV #1, EXFL
4200 012612 004737 014772 JSR PC, LTGER0 ;GO PRINT ERROR
4201 012616 000410 BR LT42X
4202 012620 012701 100200 LT42E: MOV #100200, R1 ;SET EXPT ERROR BITS
4203 012624 017702 165674 MOV @ER, R2 ;GET ERROR REGISTER
4204 012630 020102 CMP R1, R2 ;SEE IF UNEXPECTED ERRORS
4205 012632 001402 BEQ LT42X ;IF NOT: BR

```

```

4206 012634 004737 014760
4207 012640 004737 016350
4208 012644 004737 015440
4209 012650 000137 002430
4210
4211
4212
4213 012654 012737 001700 000776
4214 012662 004737 014574
4215 012666 012737 012702 000712
4216 012674 012737 025373 000622
4217 012702 004737 016516
4218 012706 005001
4219 012710 005301
4220 012712 001376
4221 012714 012777 000023 165612
4222 012722 012777 177770 165562
4223 012730 012777 177760 165560
4224 012736 012777 027314 165550
4225 012744 012777 000061 165536
4226 012752 005000
4227 012754 032777 000200 165540
4228 012762 001002
4229 012764 005300
4230 012766 001372
4231 012770 032777 000200 165526
4232 012776 001411
4233 013000 012737 027053 000672
4234 013006 012737 000002 000716
4235 013014 004737 014772
4236 013020 000425
4237 013022 017702 165506
4238 013026 042702 000177
4239 013032 012701 177600
4240 013036 020102
4241 013040 001405
4242 013042 012737 022570 000672
4243 013050 004737 016106
4244 013054 017702 165444
4245 013060 012701 000000
4246 013064 020102
4247 013066 001402
4248 013070 004737 014760
4249 013074 004737 016350
4250 013100 000137 002430

LT42X: JSR PC,LTGER3 ;ELSE PRINT ERROR
        JSR PC,ITER ;DO ITERATIONS
        JSR PC,DRVCLR
        JMP TSCD2 ;RETURN TO SCHED

;LOGIC TEST 43: LONGITUALINAL REDUNDANCY(LRC)*****
LT43: MOV #1700,UDES ;SET UNIT DESCRIPTION = NRZ
      JSR PC,STATIC ;GO SEE IF STATIC ONLY
      MOV #LT43IT,SCOLP ;SET SCOPE ADDRESS
      MOV #M5LT43,EMADDR
LT43IT: JSR PC,INIT ;INIT, SELECT DRIVE+SLAVE
        CLR R1
1$: DEC R1 ;DELAY
      BNE 1$
      MOV #23,@MR ;SET MM
      MOV #-10,@WC ;SET WC
      MOV #-20,@FC ;SET FC
      MOV #WDATA,@BA ;SET BA
      MOV #61,@C1 ;LOAD WRITE+GO
      CLR RO
LT43C: BIT #200,@DS ;SEE IF DRY
      BNE LT43D ;IF SO: BR
      RO
      BNE LT43C ;AWAIT DRY
LT43D: BIT #200,@ER ;SEE IF LRC=1
      BEQ LT43E ;IF NOT: BR
      MOV #TMS26,ERADD ;SET ERROR CODE
      MOV #2,EXFL
      JSR PC,LTGERO ;GO PRINT
      BR LT43X
LT43E: MOV @MR,R2
      BIC #177,R2 ;MASK LRC
      MOV #177600,R1 ;SET EXPT LRC
      CMP R1,R2 ;SEE IF EXPT = RCVD
      BEQ LT43F ;IF SO: BR
      MOV #MSG53,ERADD ;SET ERROR CODE
      JSR PC,LTGER1 ;PRINT ERROR
LT43F: MOV @ER,R2 ;GET ERROR REGISTER
      MOV #0,R1 ;SET EXPT ERROR BITS
      CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
      BEQ LT43X ;IF NOT: BR
      JSR PC,LTGER3 ;ELSE PRINT ERROR
LT43X: JSR PC,ITER
      JMP TSCD2 ;RETURN TO SCHED
  
```

```

4252                                     ;LOGIC TEST 4  PE CORRECTABLE DATA (CORR)*****
4253
4254 013104 012737 002300 000776 LT44:  MOV    #2300, UDES      ;SET UNIT DESRIPTION = PE
4255 013112 004737 014574          JSR    PC, STATIC      ;GO SEE IF STATIC ONLY
4256 013116 012737 025427 000622          MOV    #MSLT44, EMADDR ;SET HEADER
4257 013124 012737 013132 000712          MOV    #LT44IT, SCOLP  ;SET SCOP
4258 013132 004737 016516          LT44IT: JSR   PC, INIT   ;GO INITIALIZE
4259 013136 012777 177600 165346          MOV    #-200, @WC      ;SET WC=200
4260 013144 012777 177400 165344          MOV    #-400, @FC      ;SET FC=400
4261 013152 012777 027314 165334          MOV    #WDATA, @BA     ;SET BA=START OF WRITE BUFFER
4262 013160 012777 000061 165322          MOV    #61, @C1        ;LOAD WRITE AND GO
4263 013166 005000          CLR    R0
4264 013170 005777 165322          LT44A:  TST    @FC      ;SEE IF FC=0
4265 013174 001402          BEQ    LT44A1          ;IF SO:BR
4266 013176 005300          DEC    R0
4267 013200 001373          BNE    LT44A          ;AWAIT FC=0
4268 013202 012777 000021 165324          LT44A1: MOV    #21, @MR     ;SET MAINT MODE
4269 013210 005000          CLR    R0
4270 013212 032777 000200 165302          LT44B:  BIT    #200, @DS  ;SEE IF DRY
4271 013220 001002          BNE    LT44C          ;IF SO :BR
4272 013222 005300          DEC    R0
4273 013224 001372          BNE    LT44B          ;AWAIT DRY
4274 013226 005777 165272          LT44C:  TST    @ER      ;SEE IF CORR=1
4275 013232 100410          BMI    LT44D          ;IF SO: BR
4276 013234 012737 027254 000672          MOV    #TMS35, ERADD  ;ELSE SET ERROR CODE
4277 013242 012737 000001 000716          MOV    #1, EXFL       ;SET EXPT FLAG
4278 013250 004737 014772          JSR    PC, LTGERO     ;GO PRINT ERROR
4279 013254 000240          LT44D:  NOP
4280 013256 122777 000002 165244          LT44E:  CMPB   #2, @CC   ;SEE IF DEAD TRACK BIT 1
4281 013264 001414          BEQ    LT44F          ;IF SO: BR
4282 013266 117702 165236          MOVB  @CC, R2        ;ELSE SAVE RECVD
4283 013272 042702 177000          BIC   #177000, R2    ;MASK OUT CRC
4284 013276 112701 000002          MOVB  #2, R1         ;SAVE EXPT
4285 013302 012737 022177 000672          MOV    #MSG42, ERADD  ;SET ERROR CODE
4286 013310 004737 016106          JSR    PC, LTGER1    ;GO PRINT ERROR
4287 013314 000410          BR    LT44X
4288 013316 017702 165202          LT44F:  MOV    @ER, R2   ;GET ERROR REGISTER
4289 013322 012701 100000          MOV    #100000, R1    ;SET EXPT ERROR BITS
4290 013326 020102          CMP   R1, R2         ;SEE IF EXPT=RCVD
4291 013330 001402          BEQ   LT44X          ;IF SO: BR
4292 013332 004737 014760          JSR   PC, LTGER3     ;ELSE PRINT ERROR
4293 013336 004737 016350          LT44X:  JSR    PC, ITER ;GO SEE IF ITERATIONS
4294 013342 004737 015440          JSR    PC, DRVCLR    ;GO DO DRIVE CLEAR
4295 013346 000137 002430          LT44XX: JMP    TSCD2  ;RETURN TO SCHED

```

```

4297
4298
4299
4300 013352 012737 002300 000776 LT45: MOV #2300,UDES ;SET UNIT DESCRIPTION = PE
4301 013360 004737 014574 JSR PC,STATIC ;GO SEE IF STATIC ONLY
4302 013364 012737 025507 000622 MOV #MSLT45,EMADDR
4303 013372 012737 013400 000712 MOV #LT45IT,SCOLP
4304 013400 004737 016516 LT45IT: JSR PC,INIT ;INIT SELECT DRIVE SLAVE
4305 013404 012777 177600 165100 MOV #-200,@WC ;SET WC=200
4306 013412 012777 177400 165076 MOV #-400,@FC ;SET FC=400
4307 013420 012777 027314 165066 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
4308 013426 012777 000023 165100 MOV #23,@MR ;SET MAINT CODE
4309 013434 012777 000061 165046 MOV #61,@C1 ;LOAD WRITE+GO
4310 013442 005000 CLR R0
4311 013444 005777 165046 LT45E: TST @FC ;AWAIT FC=0
4312 013450 001402 BEQ LT45E1
4313 013452 005300 DEC R0
4314 013454 001373 BNE LT45E ;AWAIT FC=0
4315 013456 005000 LT45E1: CLR R0
4316 013460 032777 000200 165034 LT45A: BIT #200,@DS ;SEE IF DRY IS SET
4317 013466 001002 BNE LT45B ;IF SO: BR
4318 013470 005300 DEC R0
4319 013472 001372 BNE LT45A ;AWAIT DRY
4320 013474 032777 000100 165022 LT45B: BIT #100,@ER ;SEE IF INC=1
4321 013502 001010 BNE LT45D ;IF SO:BR
4322 013504 012737 027031 000672 MOV #TMS23,ERADD ;SET ERROR CODE
4323 013512 012737 000001 000716 MOV #1,EXFL
4324 013520 004737 014772 LT45D: JSR PC,LTGERO ;GO PRINT ERROR
4325 013524 017702 165000 MOV @CC,R2 ;GET CHECK CHAR
4326 013530 042702 177000 BIC #177000,R2 ;MASK CHECK CHAR
4327 013534 012701 000046 MOV #46,R1 ;SET EXPT CK
4328 013540 020102 CMP R1,R2 ;SEE IF EXPT = RCVD
4329 013542 001405 BEQ LT45F ;IF SO: BR
4330 013544 012737 027602 000672 MOV #MSG54,ERADD
4331 013552 004737 016106 JSR PC,LTGER1 ;ELSE GO PRINT ERROR
4332 013556 017702 164742 LT45F: MOV @ER,R2
4333 013562 042702 120600 BIC #120600,R2 ;MASK OPI ,NSG, CORR, AND PEF
4334 013566 012701 000100 MOV #100,R1 ;SET EXPT ERRGR BITS
4335 013572 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
4336 013574 001402 BEQ LT45X ;IF NOT: BR
4337 013576 004737 014760 JSR PC,LTGER3 ;ELSE PRINT ERROR
4338 013602 004737 016350 LT45X: JSR PC,ITER
4339 013606 004737 015440 JSR PC,DRVCLR
4340 013612 000137 002430 LT45XX: JMP TSCD2 ;RETURN TO SCHED
  
```

```

4342
4343
4344
4345 013616 012737 002300 000776 LT46:  MOV    #2300,UDES    :SET UNIT DESCRIPTION = PE
4346 013624 004737 014574          JSR    PC,STATIC    :GO SEE IF STATIC ONLY
4347 013630 012737 025571 000622          MOV    #MSLT46,EMADDR :SET HEADER
4348 013636 012737 013644 000712          MOV    #LT46IT,SCOLP  :SET SCOPE ADDRESS
4349 013644 004737 016516          LT46IT: JSR    PC,INIT    :INITIALIZE
4350 013650 012777 177770 164634          MOV    #-10,@WC      :SET WC=10
4351 013656 012777 177760 164632          MOV    #-20,@FC      :SET FC=20
4352 013664 012777 027314 164622          MOV    #WDATA,@BA    :SET BA=START OF WRITE BUFFER
4353 013672 012777 000061 164610          MOV    #61,@C1       :LOAD WRITE+GO
4354 013700 005777 164612          LT46A:  TST    @FC
4355 013704 001375          BNE    LT46A         :AWAIT FC=0
4356 013706 032777 000100 164620 1$:  BIT    #100,@MR
4357 013714 001774          BEQ    1$            ;DELAY
4358 013716 032777 000100 164610 2$:  BIT    #100,@MR
4359 013724 001374          BNE    2$
4360 013726 032777 000100 164600 3$:  BIT    #100,@MR
4361 013734 001774          BEQ    3$
4362 013736 012777 000027 164570          MOV    #27,@MR
4363 013744 005000          CLR    RO            ;SET MM CODE TO KILL PEF
4364 013746 032777 000200 164546          LT46B:  BIT    #200,@DS    :INIT TIMING LOOP
4365 013754 001002          BNE    LT46C         :SEE IF DRY SET
4366 013756 005300          DEC    RO            ;IF SO: BR
4367 013760 001372          BNE    LT46B         :AWAIT DRY
4368 013762 032777 000200 164534          LT46C:  BIT    #200,@ER    :SEE IF PEF SET
4369 013770 001011          BNE    LT46D         :IF SO: BR
4370 013772 012737 027045 000672          MOV    #TMS25,ERADD  :SET ERROR TAG
4371 014000 012737 000001 000716          MOV    #1,EXFL       :SET EXPT FLAG
4372 014006 004737 014772          JSR    PC,LTGERO     :GO PRINT ERROR
4373 014012 000412          BR     LT46X
4374 014014 017702 164504          LT46D:  MOV    @ER,R2      :GET ERROR REGISTER
4375 014020 042702 020500          BIC    #20500,R2     :CLEAR OPI BIT (MAY OR MAY NOT SET)
4376 014024 012701 000200          MOV    #200,R1       :SET EXPT ERROR BITS
4377 014030 020102          CMP    R1,R2         :SEE IF UNEXPECTED ERRORS
4378 014032 001402          BEQ    LT46X         :IF NOT: BR
4379 014034 004737 014760          JSR    PC,LTGER3     :ELSE PRINT ERROR
4380 014040 004737 016350          LT46X:  JSR    PC,ITER
4381 014044 004737 015440          JSR    PC,DRVCLR
4382 014050 000137 002430          LT46XX: JMP    TSCD2       :RETURN TO SCHED
  
```



```

4384 ;LOGIC TEST 47: FRAME COUNT OVERFLOW(M8905-YB)*****
4385
4386 014054 012737 025625 000622 LT47: MOV #MSLT47,EMADDR ;SET TEST HEADER
4387 014062 012737 014070 000712 MOV #LT47IT,SCOLP ;SET SCOPE ADDRESS
4388 014070 004737 016466 LT47IT: JSR PC,INIT$ ;GO INIT
4389 014074 012777 177770 164410 MOV #-10,@WC ;SET WC = 10
4390 014102 012777 177760 164406 MOV #-20,@FC ;SET FC = 20
4391 014110 052777 001700 164424 BIS #1700,@TC ;SET TO NRZ, NORMAL, ODD
4392 014116 012777 027314 164370 MOV #WDATA,@BA ;SET BUS ADDRESS
4393 014124 012777 000013 164402 MOV #13,@MR ;SET WRAP 2
4394 014132 012777 000061 164350 MOV #61,@C1 ;LOAD WRITE+GO
4395 014140 012700 040000 MOV #40000,R0
4396 014144 005777 164372 LT47A: TST @TC ;SEE IF ALPHA
4397 014150 100002 BPL LT47B ;IF SO: BR
4398 014152 005300 DEC R0
4399 014154 001373 BNE LT47A ;AWAIT ALPHA
4400 014156 012700 000020 LT47B: MOV #20,R0 ;SET CLK CNT
4401 014162 052777 000040 LT47C: BIS #40,@MR
4402 014170 042777 000040 164336 BIC #40,@MR ;CLOCK MR
4403 014176 005300 DEC R0
4404 014200 001370 BNE LT47C ;IF NOT DONE ALL: BR
4405 014202 017702 164310 MOV @FC,R2
4406 014206 005001 CLR R1 ;SET TEST WORD
4407 014210 020102 CMP R1,R2 ;SEE IF EXPT = RCVD
4408 014212 001410 BEQ LT47X ;IF SO: BR
4409 014214 012737 021452 000672 MOV #MSG19,ERADD ;SET ERROR CODE
4410 014222 012737 000001 000716 MOV #1,EXFL ;SET EXPT FLAG
4411 014230 004737 016106 JSR PC,LTGER1 ;GO PRINT ERROR
4412 014234 004737 016350 LT47X: JSR PC,ITER ;GO SEE IF ITERATIONS
4413 014240 000137 C02430 JMP TSCD2 ;RETURN TO SCHEDULAR
4414
  
```

```

4416
4417
4418
4419 014244 012737 025675 000622 LT50: MOV #MSLT50,EMADDR ;SET TEST HEADER
4420 014252 012737 014260 000712      MOV #LT50IT,SCOLP
4421 014260 004737 016466      LT50IT: JSR PC,INIT3 ;SET SLAVE = NRZ
4422 014264 042777 003400 164250      BIC #3400,@TC ;CLEAR DENSITY BITS
4423 014272 052777 002300 164242      BIS #2300,@TC ;SET DENSITY = PE
4424 014300 012777 177770 164204      MOV #-10,@WC ;SET WORD COUNT
4425 014306 012777 177760 164202      MOV #-20,@FC ;SET FRAME COUNT
4426 014314 012777 027314 164172      MOV #WDATA,@BA ;SET BUS ADDRESS
4427 014322 012777 000013 164204      MOV #13,@MR ;SET WRAP 2
4428 014330 012777 000061 164152      MOV #61,@C1 ;LOAD WRITE COMMAND
4429 014336 000240
4430 014340 000240      NOP
4431 014342 000240      NOP
4432 014344 012701 004000      MOV #4000,R1 ;SET EXPECTED RESULT
4433 014350 017702 164150      MOV @ER,R2 ;GET ERROR REGISTER
4434 014354 030102      BIT R1,R2 ;BRANCH IF NEF BIT SET
4435 014356 001006      BNE 1$
4436 014360 012737 000001 000716      MOV #1,EXFL ;SET EXPECTED FLAG
4437 014366 004737 014772      JSR PC,LTGERO ;PRINT ERROR
4438 014372 000404      BR LT50X
4439 014374 020102      1$: CMP R1,R2 ;BRANCH IF NO UNEXPECTED ERROR
4440 014376 001402      BEQ LT50X ;BITS WERE SET
4441 014400 004737 014760      JSR PC,LTGER3 ;PRINT ERROR MSG
4442 014404 004737 016350      LT50X: JSR PC,ITER ;ITERATE TEST
4443 014410 004737 015440      JSR PC,DRVCLR ;RESET DRIVE
4444 014414 000137 002430      JMP TSCD2
4445
4446 ;LOGIC TEST 51: NEF WHEN WRITING NRZ ON PE SELECTED SLAVE
4447 014420 012737 025755 000622 LT51: MOV #MSLT51,EMADDR ;SET ERROR MSG HEADER
4448 014426 012737 014434 000712      MOV #LT51IT,SCOLP ;SET SCOPE LOOP ADDRESS
4449 014434 004737 016502      LT51IT: JSR PC,INIT4 ;SET SLAVE = PE
4450 014440 042777 002300 164074      BIC #2300,@TC ;CLEAR DENSITY BITS
4451 014446 052777 001300 164066      BIS #1300,@TC ;SET DENSITY = NRZ
4452 014454 012777 177770 164030      MOV #-10,@WC ;SET WORD COUNT
4453 014462 012777 177760 164026      MOV #-20,@FC ;SET FRAME COUNT
4454 014470 012777 027314 164016      MOV #WDATA,@BA ;SET BUS ADDRESS
4455 014476 012777 000013 164030      MOV #13,@MR ;SET WRAP 2
4456 014504 012777 000061 163776      MOV #61,@C1 ;SET WRITE COMMAND AND GO
4457 014512 000240      NOP
4458 014514 000240      NOP
4459 014516 000240      NOP
4460 014520 012701 004000      MOV #4000,R1 ;SET EXPECTED RESULT
4461 014524 017702 163774      MOV @ER,R2 ;GET ERROR REGISTER
4462 014530 030102      BIT R1,R2 ;BRANCH IF NEF SET
4463 014532 001006      BNE 1$
4464 014534 012737 000001 000716      MOV #1,EXFL ;SET EXPECTED FLAG
4465 014542 004737 014772      JSR PC,LTGERO ;PRINT ERROR MSG
4466 014546 000404      BR LT51X
4467 014550 020102      1$: CMP R1,R2 ;BRANCH IF NO UNEXPECTED
4468 014552 001402      BEQ LT51X ;ERROR BITS WERE SET
4469 014554 004737 014760      JSR PC,LTGER3
4470 014560 004737 016350      LT51X: JSR PC,ITER ;ITERATE TEST
4471 014564 004737 015440      JSR PC,DRVCLR ;CLEAR DRIVE

```

IM03/TU45 CONTROL LOGIC TEST PART I
CZTUOB.P11 13-MAY-80 14:47

MACY11 30A(1052) 13-MAY-80^{F 7} 14:50 PAGE 75-1

SEQ 0083

4472 014570 000137 002430

JMP TSCD2

;RETURN TO SCHEDULER

4474
4475
4476 014574 005737 000740
4477 014600 001006
4478 014602 005737 001012
4479 014606 001403
4480 014610 005726
4481 014612 000137 002430
4482 014616 005037 001006
4483 014622 000207
4484

;STATIC TESTS ONLY SUBROUTINE*****

STATIC: TST STFLG ;SEE IF SINGLE TEST ONLY
BNE 1\$;IF SO: BR
TST STATC ;SEE IF STATIC ONLY
BEQ 1\$;IF NOT: BR
TST (SP)+ ;RESET STACK
JMP TSCD2 ;RETURN TO SCHEDULAR
1\$: CLR RDRVF
RTS PC ;RETURN TO TEST

```

4486
4487
4488                ;END OF RECORD FORCE SUBROUTINE*****
4489 014624 017700 163704 EORPA: MOV @MR,R0          ;GET MAINT REG
4490 014630 042700 000030 BIC #36,R0        ;CLEAR CURRENT OP CODE
4491 014634 052700 000024 BIS #24,R0         ;SET EOR CLEAR OP CODE
4492 014640 010077 163670 MOV R0,@MR          ;DO EOR
4493 014644 042777 000037 163662 BIC #37,@MR        ;CLEAR EOR AND MM
4494 014652 005000 CLR R0
4495 014654 012701 000002 MOV #2,R1
4496 014660 032777 000001 163622 EORP1: BIT #1,@C1        ;SEE IF GO GONE
4497 014666 001430 BEQ EORP2          ;IF SO: BR
4498 014670 005300 DEC R0
4499 014672 001372 BNE EORP1          ;AWAIT GO RESET
4500 014674 005301 DEC R1
4501 014676 001370 BNE EORP1
4502 014700 032777 020000 163662 BIT #20000,@SWR      ;SEE IF ERROR PRINT INHIBIT
4503 014706 001020 BNE EORP2          ;IF SO: BR
4504 014710 005737 000620 TST HDRFL          ;SEE IF DONE HEADER
4505 014714 001004 BNE EORP1A         ;IF SO: BR
4506 014716 013704 000622 MOV EMADDR,R4
4507 014722 004737 017446 JSR PC,TTOUT       ;PRINT HEADER
4508 014726 012704 027727 EORP1A: MOV #WMMSG31,R4
4509 014732 004737 017446 JSR PC,TTOUT       ;PRINT EOR GO BIT ERROR
4510 014736 032777 100000 163624 BIT #100000,@SWR   ;SEE IF HALT ON ERROR
4511 014744 001401 BEQ EORP2          ;IF NOT: BR
4512 014746 000000 HALT
4513 014750 000240 EORP2: NOP
4514 014752 005037 000676 EORPX: CLR TEMP2      ;CLEAR FLAG
4515 014756 000207 RTS PC          ;RETURN
4516

```

```

4518 ;LOGIC TEST ADDRESSING ERROR SUBROUTINE*****
4519
4520 014760 005037 000716 LTGER3: CLR EXFL
4521 014764 012737 022541 000672 MOV #MSG51,ERADD
4522 014772 012737 000001 000746 LTGER0: MOV #1,ADDFL ;SET NO ADDRESS FLAG
4523 015000 000240 LTGER: NOP
4524 015002 005037 000666 CLR PFLG ;CLEAR PRINT FLAG
4525 015006 032777 020000 163554 BIT #20000,@SWR ;SEE IF SHOULD PRINT
4526 015014 001112 LTGX: BNE LTGX ;IF NOT: BR
4527 015016 005737 000620 LTGA: TST HDRFL ;SEE IF PRINTED HEADER
4528 015022 001004 BNE LTGA1 ;IF SO: BR
4529 015024 013704 000622 MOV EMADDR,R4
4530 015030 004737 017446 JSR PC,TTOUT ;PRINT TEST HEADER
4531 015034 012737 000001 000620 LTGA1: MOV #1,HDRFL ;SET HEADER FLAG
4532 015042 013704 000672 MOV ERADD,R4
4533 015046 004737 017446 JSR PC,TTOUT ;PRINT CONDITION ERROR
4534 015052 005737 000746 TST ADDFL
4535 015056 001003 BNE LTGA2
4536 015060 010103 MOV R1,R3
4537 015062 004737 017574 JSR PC,OCTP ;PRINT ADDRESS
4538 015066 005737 000716 LTGA2: TST EXFL
4539 015072 001412 BEQ LTGC ;IF NO STATUS: BR
4540 015074 012704 021032 MOV #MSG6,R4
4541 015100 022737 000001 000716 CMP #1,EXFL ;EXPT-NOT RCVD
4542 015106 001402 BEQ LTGB
4543 015110 012704 021051 MOV #MSG7,R4 ;RCVD-NOT EXPT
4544 015114 004737 017446 LTGB: JSR PC,TTOUT ;PRINT STATUS
4545 015120 005237 000666 LTGC: INC PFLG
4546 015124 005737 000746 TST ADDFL ;SEE IF ADD TST
4547 015130 001430 BEQ LTGD ;IF SO: BR
4548 015132 005737 000744 TST T24FL ;SEE IF TEST 24
4549 015136 001423 BEQ LTGCO ;IF NOT: BR
4550 015140 012704 027714 MOV #MSG27,R4
4551 015144 004737 017446 JSR PC,TTOUT ;PRINT DATA TAG
4552 015150 012704 021352 MOV #MSG12,R4
4553 015154 004737 017446 JSR PC,TTOUT ;PRINT EXPT TAG
4554 015160 012703 177777 MOV #-1,R3
4555 015164 004737 017564 JSR PC,OCTPE ;PRINT EXPT
4556 015170 012704 021361 MOV #MSG13,R4
4557 015174 004737 017446 JSR PC,TTOUT ;PRINT RCVD TAG
4558 015200 010103 MOV R1,R3 ;GET RCVD
4559 015202 004737 017564 JSR PC,OCTPE ;PRINT RCVD
4560 015206 004737 015304 LTGCO: JSR PC,REGP ;PRINT REGISTERS
4561 015212 032777 004000 163350 LTGD: BIT #4000,@SWR
4562 015220 001010 BNE LTGX
4563 015222 012704 021423 MOV #MSG16,R4
4564 015226 004737 017446 JSR PC,TTOUT
4565 015232 013703 000702 MOV ITCNT,R3 ;PRINT ITERATION
4566 015236 004737 017574 JSR PC,OCTP
4567 015242 005777 163322 LTGX: TST @SWR
4568 015246 100001 BPL LTGXA ;IF NOT STOP ON ERROR: BR
4569 015250 000000 HALT
4570 015252 005737 000666 LTGXA: TST PFLG
4571 015256 001004 BNE LTGXX ;IF PRINTED: BR
4572 015260 032777 020000 163302 BIT #20000,@SWR
4573 015266 001653 BEQ LTGA

```

4574 015270 005037 000746
 4575 015274 005037 000716
 4576 015300 000137 016320
 4577
 4578
 4579
 4580 015304 000240
 4581 015306 012704 022364
 4582 015312 004737 017446
 4583 015316 017703 163166
 4584 015322 004737 017564
 4585 015326 017703 163160
 4586 015332 004737 017564
 4587 015336 017703 163152
 4588 015342 004737 017564
 4589 015346 017703 163144
 4590 015352 004737 017564
 4591 015356 017703 163136
 4592 015362 004737 017564
 4593 015366 017703 163130
 4594 015372 004737 017564
 4595 015376 017703 163122
 4596 015402 004737 017564
 4597 015406 017703 163114
 4598 015412 004737 017564
 4599 015416 017703 163112
 4600 015422 004737 017564
 4601 015426 017703 163110
 4602 015432 004737 017564
 4603 015436 000207
 4604
 4605

LTGXX: CLR ADDFL ;CLEAR ADDRESS FLAG
 CLR EXFL
 JMP SCOPE
 ;SUBROUTINE TO PRINT MAJOR REGISTERS*****
 REGP: NOP
 MOV #MSG46,R4
 JSR PC,TIOUT ;PRINT REGISTER HEADER
 MOV @C1,R3
 JSR PC,OC TPE
 MOV @WC,R3
 JSR PC,OC TPE
 MOV @BA,R3
 JSR PC,OC TPE
 MOV @FC,R3
 JSR PC,OC TPE
 MOV @CS,R3
 JSR PC,OC TPE
 MOV @DS,R3
 JSR PC,OC TPE ;PRINT REGISTERS
 MOV @ER,R3
 JSR PC,OC TPE
 MOV @AS,R3
 JSR PC,OC TPE
 MOV @MR,R3
 JSR PC,OC TPE
 MOV @TC,R3
 JSR PC,OC TPE
 RTS PC

```

4607                                     ;DRIVE CLEAR SUBROUTINE*****
4608
4609 015440 000240          DRVCLR: NOP
4610 015442 012704 040000      MOV      #40000,R4
4611 015446 005304          DCD:    DEC      R4
4612 015450 001376          BNE     DCD           ;DELAY
4613 015452 005037 000666      CLR     PFLG
4614 015456 004737 015674      JSR     PC,ATTN      ;GO SEE OF ATTN SET
4615 015462 012777 000011 163020  MOV     #11,@C1      ;ISSUE DRIVE CLEAR
4616 015470 005000          CLR     R0
4617 015472 032777 000200 163022  DCA:    BIT     #200,@DS      ;SEE IF DRY
4618 015500 001002          BNE     DCA0
4619 015502 005300          DEC     R0
4620 015504 001372          BNE     DCA           ;WAIT FOR DRY
4621 015506 032777 040000 163006  DCA0:   BIT     #40000,@DS      ;SEE IF ERR RESET
4622 015514 001022          BNE     DCE           ;IF NOT: BR
4623 015516 005777 163002      TST     @ER           ;SEE IF ERROR REGISTER RESET
4624 015522 001017          BNE     DCE           ;IF NOT: BR
4625 015524 005777 162772      TST     @DS           ;SEE IF ATA RESET
4626 015530 100414          BMI     DCE           ;IF NOT: BR
4627 015532 012703 000001      MOV     #1,R3         ;SET TEST BIT
4628 015536 013704 000624      MOV     DRVN,R4       ;GET DRIVE NUMBER & BRANCH
4629 015542 001403          BEQ     DCC           ;IF DRIVE 0
4630 015544 006303          DCB:    ASL     R3         ;POSITION TEST BIT PER DRIVE NUMBER
4631 015546 005304          DEC     R4
4632 015550 001375          BNE     DCB           ;IF NOT: BR
4633 015552 030377 162750      DCC:    BIT     R3,@AS      ;SEE IF ATTEN IS RESET
4634 015556 001001          BNE     DCE           ;IF NOT: BR
4635 015560 000207          RTS     PC           ;RETURN
4636
4637 015562 000240          DCE:    NOP
4638 015564 032777 020000 162776  BIT     #20000,@SWR      ;SEE IF ERROR PRINT INHIBIT
4639 015572 001017          BNE     DCEX          ;IF SO: BR
4640 015574 005737 000620      TST     HDRFL         ;SEE IF PRINT HEADER
4641 015600 001004          BNE     DCEA          ;IF NOT: BR
4642 015602 013704 000622      MOV     EMADDR,R4
4643 015606 004737 017446      JSR     PC,TTOUT       ;PRINT HEADER
4644 015612 012704 022470      DCEA:   MOV     #MSG47,R4
4645 015616 004737 017446      JSR     PC,TTOUT       ;PRINT DRIVE CLEAR ERROR
4646 015622 004737 015304      JSR     PC,REGP        ;PRINT REGISTERS
4647 015626 005237 000666      INC     PFLG          ;SET PRINTED FLAG
4648 015632 005777 162732      DCEX:   TST     @SWR      ;SEE IF HALT ON ERROR
4649 015636 100001          BPL     DCEXA         ;IF NOT: BR
4650 015640 000000          HALT
4651 015642 005737 000666      DCEXA:  TST     PFLG          ;SEE IF HAVE PRINTED
4652 015646 001004          BNE     DCEXX         ;IF SO: BR
4653 015650 032777 020000 162712  BIT     #20000,@SWR      ;BRANCH IF ERROR
4654 015656 001741          BEQ     DCE           ;PRINTOUT DESIRED
4655 015660 000240          DCEXX:  NOP
4656 015662 012737 015440 000712  MOV     #DRVCLR,SCOLP   ;SET SCOPE LOOP ADDRESS
4657 015670 000137 016320      JMP     SCOPE          ;GO DO SCOPE LOOP
  
```



```

4659 ;COMPOSITE ERROR CHECK SUBROUTINE*****
4660
4661 015674 000240 ATTN: NOP
4662 015676 005777 162620 TST @DS ;SEE IF ATA SET
4663 015702 001004 BNE ATTA ;IF SO: BR
4664 015704 012737 022026 000700 MOV #MSG32,TEMP3
4665 015712 000427 BR ATTP ;ELSE PRINT ERROR
4666 015714 032777 040000 162600 ATTA: BIT #40000,@DS ;SEE IF COMPOSITE ERROR SET
4667 015722 001004 BNE ATTB ;IF SO: BR
4668 015724 012737 022010 000700 MOV #MSG31,TEMP3
4669 015732 000417 BR ATTP ;ELSE PRINT ERROR
4670 015734 012703 000001 ATTB: MOV #1,R3 ;SET TEST BIT
4671 015740 012737 022044 000700 MOV #MSG33,TEMP3
4672 015746 013704 000624 MOV DRVN,R4 ;GET DRIVE NUMBER & BRANCH
4673 015752 001403 BEQ ATTD ;IF DRIVE 0
4674 015754 006303 ATTC: ASL R3 ;POSITION TEST BIT
4675 015756 005304 DEC R4 ;SEE IF DONE
4676 015760 001375 BNE ATTC ;IF NOT: BR
4677 015762 030377 162540 ATTD: BIT R3,@AS ;SEE IF ATTN SUMMARY SET
4678 015766 001401 BEQ ATTP ;IF NOT: BR
4679 015770 000207 RTS PC ;ELSE RETURN
4680 015772 032777 020000 162570 ATTP: BIT #20000,@SWR ;SEE IF PRINT INHIBIT
4681 016000 001021 BNE ATTX ;IF SO: BR
4682 016002 005737 000620 TST HDRFL ;SEE IF DONE HEADER
4683 016006 001004 BNE ATTPA ;IF SO: BR
4684 016010 013704 000622 MOV EMADDR,R4
4685 016014 004737 017446 JSR PC,TTOUT ;PRINT HEADER
4686 016020 013704 000700 ATTPA: MOV TEMP3,R4
4687 016024 004737 017446 JSR PC,TTOUT ;PRINT ERROR TYPE
4688 016030 004737 015304 JSR PC,REGP ;PRINT REGISTERS
4689 016034 005237 000666 INC PFLG ;SET PRINT FLAG
4690 016040 005237 000620 INC HDRFL ;SET HEADER FLAG
4691 016044 005777 162520 ATTX: TST @SWR ;SEE IF HALT ON ERROR
4692 016050 100001 BPL ATTXA ;IF NOT: BR
4693 016052 000000 HALT
4694 016054 005737 000666 ATTXA: TST PFLG ;SEE IF DONE PRINT
4695 016060 001004 BNE ATTX ;IF SO: BR
4696 016062 032777 020000 162500 BIT #20000,@SWR ;BRANCH IF NO ERROR
4697 016070 001740 BEQ ATTP ;PRINTOUT DESIRED
4698 016072 005037 000666 ATTXX: CLR PFLG ;CLEAR PRINT FLAG
4699 016076 000207 RTS PC ;RETURN
  
```

```

4701                                     ;LOGIC TEST REGISTER BIT ERROR SUBROUTINE*****
4702
4703 016100 012737 000001 000736 LTGER2: MOV    #1,PEXFL      ;SET FLAG
4704 016106 000240                LTGER1: NOP
4705 016110 005037 000666                CLR    PFLG          ;CLEAR PRINT FLAG
4706 016114 032777 020000 162446        BIT    #20000,@SWR   ;BRANCH IF ERROR
4707 016122 001055                BNE   LTG1X          ;PRINTOUT DESIRED
4708 016124 005737 000620                LTG1A: TST   HDRFL   ;SEE IF PRINT HEADER
4709 016130 001004                BNE   LTG1B          ;IF NOT: BR
4710 016132 013704 000622                MOV   EMADDR,R4
4711 016136 004737 017446                JSR   PC,TTOUT      ;PRINT HEADER
4712 016142 012737 000001 000620 LTG1B: MOV   #1,HDRFL   ;SET FLAG
4713 016150 013704 000672                MOV   ERADD,R4
4714 016154 004737 017446                JSR   PC,TTOUT      ;PRINT ERROR CODE
4715 016160 005737 000736                TST   PEXFL         ;SEE IF PRINT EXPT-RCVD
4716 016164 001016                BNE   LTG1T          ;IF NOT: BR
4717 016166 012704 021352                MOV   #MSG12,R4
4718 016172 004737 017446                JSR   PC,TTOUT      ;PRINT EXPT TAG
4719 016176 010103                MOV   R1,R3
4720 016200 004737 017574                JSR   PC,OCTP       ;PRINT EXPT
4721 016204 012704 021361                MOV   #MSG13,R4
4722 016210 004737 017446                JSR   PC,TTOUT      ;PRINT RCVD TAG
4723 016214 010203                MOV   R2,R3
4724 016216 004737 017574                JSR   PC,OCTP       ;PRINT RCVD
4725 016222 032777 004000 162340 LTG1T: BIT    #4000,@SWR
4726 016230 001010                BNE   LTG1C
4727 016232 012704 021423                MOV   #MSG16,R4
4728 016236 004737 017446                JSR   PC,TTOUT
4729 016242 013703 000702                MOV   ITCNT,R3
4730 016246 004737 017574                JSR   PC,OCTP       ;PRINT ITERATION
4731 016252 005237 000666                LTG1C: INC    PFLG
4732 016256 000240                LTG1X: NOP
4733 016260 005777 162304                TST   @SWR
4734 016264 100001                BPL   LTG1X1        ;IF NOT STOP ON ERROR: BR
4735 016266 000000                HALT
4736 016270 005737 000666                LTG1X1: TST   PFLG
4737 016274 001004                BNE   LTG1XX        ;IF HAVE PRINTED: BR
4738 016276 032777 020000 162264        BIT    #20000,@SWR
4739 016304 001707                BEQ   LTG1A
4740 016306 000240                LTG1XX: NOP
4741 016310 005037 000736                CLR   PEXFL         ;CLEAR EXPT-RCVD FLAG
4742 016314 000137 016320                JMP   SCOPE         ;GO TO SCOPE
4743
4744
4745                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
4746
4747 016320 000240                SCOPE: NOP
4748 016322 032777 040000 162240        BIT    #40000,@SWR  ;SEE IF LOOP ON ERROR
4749 016330 001001                BNE   1$           ;IF SO: BR
4750 016332 000207                RTS   PC            ;ELSE EXIT
4751 016334 000240                1$:  NOP
4752 016336 005726                TST   (SP)+         ;RESET STACK
4753 016340 000240                NOP
4754 016342 000240                NOP
4755 016344 000177 162342                JMP   @SCOLP        ;LOOP ON ERROR
4756

```

4757

;TEST ITERATION SUBROUTINE*****

4758

4759 016350 032777 004000 162212 ITER:

BIT #4000,@SWR ;SEE IF ITERATIONS

4760 016356 001403

BEQ 2\$;IF SO: BR

4761 016360 005037 000702 1\$:

CLR ITCNT ;CLEAR ITERATION COUNTER

4762 016364 000207

RTS PC ;ELSE EXIT

4763 016366 005737 001016 2\$:

TST PCNTR ;NO SUBTEST ITERATIONS ON FIRST PASS

4764 016372 001772

BEQ 1\$

4765 016374 005237 000702

INC ITCNT ;BUMP COUNTER

4766 016400 023737 000702 000606

CMP ITCNT,ITAMT ;SEE IF DONE ALL

4767 016406 001764

BEQ 1\$;IF SO: BR

4768 016410 005726

TST (SP)+ ;RESET STACK

4769 016412 017700 162276

MOV @ITRLP,R0 ;SET ITERATION POINTER

4770 016416 000110

JMP (R0) ;GO ITERATE

4771

;MANUAL INTERVENTION INHIBIT*****

4772

4773

4774 016420 000240

INMT: NOP

4775 016422 012704 022214

MOV #MSG43,R4

4776 016426 004737 017446

JSR PC,TTOUT ;GO PRINT INHIB MSG

4777 016432 000000

HALT

4778 016434 000137 002430

JMP TSCD2 ;RETURN TO SCHED

4779

```

4781
4782
4783
4784 016440 000240
4785 016442 012777 000040 162050
4786 016450 013777 000624 162042
4787 016456 013777 000664 162056
4788 016464 000207
4789
4790
4791
4792
4793
4794
4795
4796 016466 013746 000776
4797 016472 012737 001400 000776
4798 016500 000410
4799
4800
4801 016502 013746 000776
4802 016506 012737 002000 000776
4803 016514 000402
4804
4805
4806
4807 016516 013746 000776
4808 016522 012777 000040 161770
4809 016530 013777 000624 161762
4810 016536 013777 000664 161776
4811 016544 013746 000776
4812 016550 042716 174377
4813 016554 022726 001400
4814 016560 001005
4815 016562 032777 000040 161732
4816 016570 001420
4817 016572 000404
4818 016574 032777 000040 161720 1$:
4819 016602 001013
4820 016604 012777 000007 161676 2$:
4821 016612 032777 000200 161702 20$:
4822 016620 001774
4823 016622 032777 020000 161672 3$:
4824 016630 001374
4825 016632 053777 000776 161702 4$:
4826 016640 032777 000002 161654
4827 016646 001407
4828 016650 012777 000025 161632
4829 016656 032777 000200 161636 5$:
4830 016664 001774
4831 016666 012777 000011 161614 6$:
4832 016674 012637 000776
4833 016700 000207
4834
4835
4836

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

;ROUTINES TO INITIALIZE SLAVE. THESE ROUTINES PLACE THE SLAVE
;IN PROPER STATUS FOR THE CALLING TEST. INIT3 PLACES THE SLAVE IN
;NRZ MODE AND OFF BOT; INIT4 PLACES THE SLAVE IN PE MODE AND OFF
;BOT. IF THE SLAVE IS IN THE PROPER STATUS ON ENTRY NO ACTION IS TAKEN.

;SET SLAVE IN NRZ OFF BOT
INIT3: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
      MOV #1400,UDES ;SET UNIT DESCRIPTION = NRZ
      BR INITS ;GO TO INITS ROUTINE

;SET SLAVE IN PE OFF BOT
INIT4: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
      MOV #2000,UDES ;SET UNIT DESCRIPTION = PE
      BR INITS ;GO DO IT

;THIS ROUTINE IS ENTERED AT INIT WHEN THE CALLER HAS SETUP UDES.
;IT IS ENTERED AT INITS WHEN EITHER INIT3 OR INIT4 HAS SET UP UDES.
INIT: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
INIT5: MOV #40,@CS ;INIT CONTROLLER
      MOV DRVN,@CS ;SELECT TM03 DRIVE
      MOV SLVN,@TC ;SELECT TU45 SLAVE
      MOV UDES,-(SP) ;GET SLAVE DESCRIPTION
      BIC #174377,(SP) ;CLEAR ALL BUT DENSITY SELECT BITS
      CMP #1400,(SP)+ ;BRANCH IF REQUESTING PE MODE
      BNE 1$
      BIT #40,@DS ;BRANCH IF SLAVE IS IN NRZ MODE
      BEQ 4$ ;(PES = 0)
      BR 2$
      BIT #40,@DS ;BRANCH IF SLAVE IS IN PE MODE
      BNE 4$
      MOV #7,@C1 ;REWIND SLAVE
      BIT #200,@DS ;WAIT FOR READY
      BEQ 20$
      BIT #20000,@DS ;WAIT UNTIL PIP CLEARS
      BNE 3$
      BIS UDES,@TC ;LOAD SLAVE DESCRIPTION
      BIT #2,@DS ;BRANCH IF NOT AT BOT
      BEQ 6$
      MOV #25,@C1 ;ERASE TO GET OFF BOT
      BIT #200,@DS ;WAIT FOR READY
      BEQ 5$
      MOV #11,@C1 ;RESET DRIVE
      MOV (SP)+,UDES ;RESTORE UNIT DESCRIPTION
      RTS PC ;RETURN
  
```

```

4837                                     ;MANUAL INSTRUCTION SUBROUTINE*****
4838
4839 016702 000240
4840 016704 004737 017446 INST: NOP
4841 016710 012704 026035 JSR PC,TTOUT ;PRINT INSTRUCTION
4842 016714 004737 017446 MOV #MSG0,R4
4843 016720 012705 000700 JSR PC,TTOUT ;PRINT REPLY
4844 016724 012701 000001 MOV #TEMP3,R5
4845 016730 012702 177777 MOV #1,R1
4846 016734 012703 000000 MOV #-1,R2
4847 016740 004737 017106 JSR PC,TTR ;AWAIT REPLY
4848 016744 000240 NOP
4849 016746 000207 RTS PC ;EXIT
4850
4851                                     ;MAG TAPE INTERRUPT HANDLER*****
4852
4853 016750 000240 MTINT: NOP
4854 016752 013716 000670 MOV RTRN,(SP) ;SET RETURN FROM INTERRUPT ADDRESS
4855 016756 000002 RTI ;RETURN
4856
4857                                     ;TTY INTERRUPT HANDLER*****
4858
4859 016760 017746 161610 TTINT: MOV @TKB,-(SP) ;GET CHARACTER
4860 016764 042716 000200 BIC #200,(SP) ;CLEAR PARITY BIT
4861 016770 122716 000003 CMPB #3,(SP) ;BRANCH IF NOT CONTROL C
4862 016774 001010 BNE 1$
4863 016776 005737 001416 TST CHNFLG ;INHIBIT ^C IF IN CHAIN MODE
4864 017002 001005 BNE 1$
4865 017004 005077 161556 CLR @PSW ;CLEAR PSW
4866 017010 000005 RESET
4867 017012 000137 000200 JMP @#200 ;RESTART
4868 017016 122716 000001 1$: CMPB #1,(SP) ;BRANCH IF NOT ^A
4869 017022 001017 BNE 2$
4870 017024 022737 000176 000570 CMP #SWREG,SWR ;BRANCH IF USING HARDWARE SWR
4871 017032 001016 BNE 3$
4872 017034 012737 177570 000570 MOV #177570,SWR ;INVOKE HARDWARE SWR
4873 017042 004737 020320 JSR PC,.SAVE ;SAVE REGISTERS ON THE STACK
4874 017046 012704 023016 MOV #MSC63,R4 ;TYPE 'HARDWARE SWR IN USE'
4875 017052 004737 017446 JSR PC,TTOUT
4876 017056 004737 020342 JSR PC,.RESTORE
4877 017062 122716 000007 2$: CMPB #7,(SP) ;BRANCH IF NOT ^G
4878 017066 001005 BNE 4$
4879 017070 012737 000176 000570 3$: MOV #SWREG,SWR ;INVOKE SOFTWARE SWR
4880 017076 004737 020222 JSR PC,GTSWR ;GET SWITCHES
4881 017102 005726 4$: TST (SP)+ ;POP CHARACTER OFF STACK
4882 017104 000002 RTI ;RETURN
4883

```

```

4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902 017106 010146
4903 017110 011601
4904 017112 005037 000674
4905 017116 005000
4906 017120 004737 017366
4907 017124 122737 000003 000616
4908 017132 001003
4909 017134 000005
4910 017136 000137 000200
4911 017142 122737 000015 000616
4912 017150 001004
4913 017152 005737 000674
4914 017156 001471
4915 017160 000457
4916 017162 122737 000025 000616
4917 017170 001005
4918 017172 012704 022744
4919 017176 004737 017446
4920 017202 000742
4921 017204 122737 000177 000616
4922 017212 001012
4923 017214 000241
4924 017216 006000
4925 017220 006200
4926 017222 006200
4927 017224 012704 022746
4928 017230 004737 017446
4929 017234 005201
4930 017236 000730
4931 017240 122737 000060 000616
4932 017246 101402
4933 017250 000137 017346
4934 017254 122737 000070 000616
4935 017262 101002
4936 017264 000137 017346
4937 017270 005237 000674
4938 017274 006300
4939 017276 006300
4940 017300 006300

```

```

:*****
:TTY ENTRY SUBROUTINE:
:
:THIS SUBROUTINE IS USED BY THE TEST CONDITION
:ENTRY ROUTINE TO READ THE RESPONSE ENTERED
:AT THE TTY AND CHECK THEM FOR LEGALITY AND
:LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
:(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
:THE CALLING ROUTINE.
:IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
:A QUESTION MARK IS TYPED (?) AND THE RESPONSE
:MAY BE REENTERED.
:ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
:MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
:CARRIAGE RETURN
:*****
TTR: MOV R1, -(SP) ;SAVE CHARACTER COUNT
10$: MOV (SP), R1 ;RESTORE CHARACTER COUNT (FOR ^U)
CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
CLR R0
1$: JSR PC, TTIN ;GO READ CHARACTER
CMPB #3, TIB ;BRANCH IF NOT ^C
BNE 11$
RESET ;RESET
JMP @#200 ;RESTART PROGRAM
11$: CMPB #15, TIB ;SEE IF CR
BNE 2$ ;IF NOT: BR
TST TEMP1 ;SEE IF FIRST CHARACTER
BEQ 9$ ;IF SO: BR
BR 6$
2$: CMPB #25, TIB ;BRANCH IF NOT CONTROL U
BNE 21$
MOV #MSG59, R4 ;TYPE <CR><LF>
JSR PC, TTOUT
BR 10$
21$: CMPB #177, TIB ;BRANCH IF NOT 'RUBOUT'
BNE 3$
CLC
ROR R0 ;REMOVE LAST CHAR
ASR R0
ASR R0
MOV #MSG60, R4 ;TYPE '\ '
JSR PC, TTOUT
INC R1 ;DECREMENT CHARS RECEIVED COUNT
BR 1$
3$: CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0
BLOS 4$ ;IF NOT: BR
JMP TIB ;ELSE GO TO ERROR
4$: CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7
BHI 5$ ;IF NOT: BR
JMP TIB ;ELSE GO TO ERROR
5$: INC TEMP1 ;SET FIRST CHARACTER FLAG
ASL R0
ASL R0 ;SHIFT 3 LEFT
ASL R0

```

4941	017302	042737	177770	000616	BIC	#177770,T1B	:STRIP ASCII
4942	017310	053700	000616		BIS	T1B,R0	:LOAD CHARACTER
4943	017314	005301			DEC	R1	:SEE IF DONE
4944	017316	001300			BNE	1\$:IF NOT: BR
4945	017320	020002		6\$:	CMP	R0,R2	:SEE IF EXCEEDED MAXIMUM LIMIT
4946	017322	101402			BLOS	7\$:IF NOT: BR
4947	017324	000137	017346		JMP	T1NER	:ELSE GO TO ERROR
4948	017330	020300		7\$:	CMP	R3,R0	:SEE IF BELOW MINIMUM LIMIT
4949	017332	101402			BLOS	8\$:IF NOT: BR
4950	017334	000137	017346		JMP	T1NER	:ELSE GO TO ERROR
4951	017340	010015		8\$:	MOV	R0,(R5)	:LOAD VALUE
4952	017342	005726		9\$:	TST	(SP)+	:POP CHAR COUNT OFF STACK
4953	017344	000207			RTS	PC	:EXIT
4954							

```

4956
4957
4958
4959 017346 012704 022154
4960 017352 004737 017446
4961 017356 005726
4962 017360 162716 000020
4963 017364 000207
4964
4965
4966
4967 017366 017746 161174
4968 017372 012777 000340 161166
4969 017400 005277 161166
4970 017404 105777 161162
4971 017410 100375
4972 017412 012677 161150
4973 017416 017737 161152 000616
4974 017424 042737 000200 000616
4975 017432 013737 000616 000614
4976 017440 004737 017546
4977 017444 000207
4978
4979
4980
4981 017446 112437 000614
4982 017452 122737 000043 000614
4983 017460 001440
4984 017462 122737 000045 000614
4985 017470 001403
4986 017472 004737 017546
4987 017476 000763
4988 017500 112737 000015 000614
4989 017506 004737 017546
4990 017512 012703 000004
4991 017516 005037 000614
4992 017522 004737 017546
4993 017526 005303
4994 017530 001372
4995 017532 112737 000012 000614
4996 017540 004737 017546
4997 017544 000740
4998 017546 105777 161024
4999 017552 100375
5000 017554 113777 000614 161016
5001 017562 000207
5002
5003
5004
5005
5006 017564 012737 000001 020014
5007 017572 000402
5008 017574 005037 020014
5009 017600 010304
5010 017602 001006
5011 017604 005737 020014

```

```

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

;TTY READ SUBROUTINE*****
TTIN: MOV @PSW,-(SP) ;SAVE CURRANT STATUS
MOV #340,@PSW ;SET STATUS TO LEVEL 7
INC @TKS
1$: TSTB @TKS
BPL 1$
MOV (SP)+,@PSW ;RESTORE ORIGINAL STATUS
MOV @TKB,TIB
BIC #200,TIB ;STRIP PARITY BIT
MOV TIB,TOB ;MOVE CHAR TO TTY OUTPUT BFR
JSR PC,TOG ;AND ECHO IT
RTS PC

;TTY OUTPUT SUBROUTINE*****
TTOUT: MOVB (R4)+,TOB
CMPB #43,TOB
BEQ TEX
CMPB #45,TOB
BEQ 1$
JSR PC,TOG
BR TTOUT
1$: MOVB #15,TOB
JSR PC,TOG
MOV #4,R3
2$: CLR TOB
JSR PC,TOG
DEC R3
BNE 2$ ;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
BNF OCTP0 ;IF NOT ZERO: BR
TSI OFL ;SEE IF PRINT ALL 0

```


5012	017610	001003			BNE	OCTP0		:IF SO: BR
5013	017612	004737	017774		JSR	PC,OCTPG1		:ELSE PRINT ZERO
5014	017616	000447			BR	OCTP3		:SPACE AND EXIT
5015	017620	032704	100000		OCTP0: BIT	#100000,R4		:SEE IF MSD = 1
5016	017624	001405			BEO	OCTP1		:IF NOT: BR
5017	017626	012704	000001		MOV	#1,R4		
5018	017632	004737	017752		JSR	PC,OCTPG		:PRINT 1
5019	017636	000403			BR	OCTP2		
5020	017640	005004			OCTP1: CLR	R4		
5021	017642	004737	017752		JSR	PC,OCTPG		:PRINT 0
5022	017646	010304			OCTP2: MOV	R3,R4		
5023	017650	006004			ROR	R4		
5024	017652	006004			ROR	R4		
5025	017654	006004			ROR	R4		:POSITION DIGIT
5026	017656	006004			ROR	R4		
5027	017660	000304			SWAB	R4		
5028	017662	004737	017752		JSR	PC,OCTPG		:PRINT DIGIT 2
5029	017666	010304			MOV	R3,R4		
5030	017670	006004			ROR	R4		
5031	017672	000304			SWAB	R4		
5032	017674	004737	017752		JSR	PC,OCTPG		:PRINT DIGIT 3
5033	017700	010304			MOV	R3,R4		
5034	017702	006104			ROL	R4		
5035	017704	006104			ROL	R4		
5036	017706	000304			SWAB	R4		
5037	017710	004737	017752		JSR	PC,OCTPG		:PRINT DIGIT 4
5038	017714	010304			MOV	R3,R4		
5039	017716	006004			ROR	R4		
5040	017720	006004			ROR	R4		
5041	017722	006004			ROR	R4		
5042	017724	004737	017752		JSR	PC,OCTPG		
5043	017730	010304			MOV	R3,R4		
5044	017732	004737	017752		JSR	PC,OCTPG		:PRINT DIGIT 5
5045	017736	012737	000240	000614	OCTP3: MOV	#240,TOB		
5046	017744	004737	017546		JSR	PC,TOG		:PRINT SPACE
5047	017750	000207			RTS	PC		:EXIT
5048								
5049	017752	042704	177770		OCTPG: BIC	#177770,R4		
5050	017756	001004			BNE	OCTPG0		
5051	017760	005737	020014		TST	OFL		
5052	017764	001001			BNE	OCTPG0		
5053	017766	000207			RTS	PC		
5054								
5055	017770	005237	020014		OCTPG0: INC	OFL		
5056	017774	052704	000260		OCTPG1: BIS	#260,R4		
5057	020000	010437	000614		MOV	R4,TOB		
5058	020004	004737	017546		JSR	PC,TOG		
5059	020010	010304			MOV	R3,R4		
5060	020012	000207			RTS	PC		
5061	020014	000000			OFL: 0			:FIRST CHAR FLAG
5062								

```

5064
5065                                     ;DATA CHARACTER OUTPUT SUBROUTINE*****
5066
5067 020016 012704 000010          DOUT:  MOV    #10,R4          ;SET NUMBER TO PRINT
5068 020022 110337 000614          MOVB   R3,TOB
5069 020026 105777 160544          1$:   TSTB   @TPS
5070 020032 100375                    BPL    1$
5071 020034 132737 000200 000614  BITB   #200,TOB
5072 020042 001404                    BEQ    2$
5073 020044 012777 000061 160526  MOV    #061,@TPB
5074 020052 000403                    BR     3$
5075 020054 012777 000060 160516  2$:   MOV    #060,@TPB
5076 020062 006337 000614          3$:   ASL    TOB
5077 020066 005304                    DEC    R4
5078 020070 001356                    BNE   1$
5079 020072 000207                    RTS    PC
5080
5081 020074 013703 000700          DOUTD: MOV    TEMP3,R3
5082 020100 000303                    SWAB  R3
5083 020102 004737 020016          JSR   PC,DOUT
5084 020106 013703 000700          MOV   TEMP3,R3
5085 020112 004737 020016          JSR   PC,DOUT
5086 020116 000207                    RTS    PC
5087
5088                                     ;TU45 SERIAL NUMBER PRINT SUBROUTINE*****
5089
5090 020120 010304          SNPT:  MOV    R3,R4
5091 020122 000304          SWAB  R4
5092 020124 006004          ROR   R4
5093 020126 006004          ROR   R4
5094 020130 006004          ROR   R4
5095 020132 006004          ROR   R4
5096 020134 004737 020176  JSR   PC,SNPG          ;GET FIRST DIGIT
5097 020140 010304          MOV   R3,R4          ;PRINT
5098 020142 000304          SWAB  R4
5099 020144 004737 020176  JSR   PC,SNPG          ;GET SECOND DIGIT
5100 020150 010304          MOV   R3,R4          ;PRINT
5101 020152 006004          ROR   R4
5102 020154 006004          ROR   R4
5103 020156 006004          ROR   R4
5104 020160 006004          ROR   R4
5105 020162 004737 020176  JSR   PC,SNPG          ;PRINT THIRD DIGIT
5106 020166 010304          MOV   R3,R4
5107 020170 004737 020176  JSR   PC,SNPG          ;PRINT FOURTH DIGIT
5108 020174 000207          RTS    PC          ;EXIT
5109 020176 012737 000260 000614  SNPG:  MOV    #260,TOB          ;SET BASE = 0
5110 020204 042704 177760          BIC   #177760,R4      ;MASK DIGIT
5111 020210 050437 000614          BIS   R4,TOB          ;SET ASCII
5112 020214 004737 017546          JSR   PC,TOG          ;TYPE DIGIT
5113 020220 000207          RTS    PC          ;RETURN

```

```

5115
5116 ;ROUTINE TO LOAD CONTENTS OF SOFTWARE SWITCH REGISTER.
5117 ;IF A CONTROL G (^G) IS TYPED THE SOFTWARE SWITCH REGISTER IS LOADED
5118 020222 022737 000176 000570 GTSWR: CMP #SWREG,SWR ;BRANCH IF SOFTWARE SWR
5119 020230 001032 BNE 1$ ;NOT INVOKED
5120 020232 004737 020320 JSR PC,SAVE ;SAVE REGISTERS ON THE STACK
5121 020236 012704 026603 MOV #SMSWR,R4 ;TYPE 'SWR = '
5122 020242 004737 017446 JSR PC,TTOUT
5123 020246 017703 160316 MOV @SWR,R3 ;GET CURRENT VALUE
5124 020252 004737 017564 JSR PC,OCTPE ;AND TYPE IT
5125 020256 012704 026613 MOV #SMNEW,R4 ;ASK FOR NEW VALUE
5126 020262 004737 017446 JSR PC,TTOUT
5127 020266 013705 000570 MOV SWR,R5 ;NEW VALUE WILL BE RETURNED IN (R5)
5128 020272 012701 000007 MOV #7,R1 ;LIMIT TO 7 CHARACTERS
5129 020276 012702 177777 MOV #177777,R2 ;LIMIT RESPONSE TO BETWEEN
5130 020302 012703 000000 MOV #0,R3 ;0 AND 177777
5131 020306 004737 017106 JSR PC,TTR ;GET RESPONSE
5132 020312 004737 020342 JSR PC,.RESTORE ;RESTORE REGISTERS
5133 020316 000207 1$: RTS PC ;RETURN TO CALLER
5134 ;:ROUTINE TO SAVE REGISTERS ON THE STACK
(1) 020320 010546 .SAVE: MOV %5,-(SP) ;;R5 IS SAVED AT 12(SP)
(1) 020322 010446 MOV %4,-(SP) ;;R4 IS SAVED AT 10(SP)
(1) 020324 010346 MOV %3,-(SP) ;;R3 IS SAVED AT 6(SP)
(1) 020326 010246 MOV %2,-(SP) ;;R2 IS SAVED AT 4(SP)
(1) 020330 010146 MOV %1,-(SP) ;;R1 IS SAVED AT 2(SP)
(1) 020332 010046 MOV %0,-(SP) ;;R0 IS SAVED AT (SP)
(1) 020334 016646 000014 MOV 14(SP),-(SP) ;;PUSH RETURN PC ON THE STACK
(1) 020340 000207 RTS PC ;;RETURN TO CALLER
5135 ;:ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
(1) 020342 012666 000014 .RESTORE:MOV (SP)+,14(SP) ;;STORE RETURN PC ON STACK
(1) 020346 012600 MOV (SP)+,%0
(1) 020350 012601 MOV (SP)+,%1
(1) 020352 012602 MOV (SP)+,%2
(1) 020354 012603 MOV (SP)+,%3
(1) 020356 012604 MOV (SP)+,%4
(1) 020360 012605 MOV (SP)+,%5
(1) 020362 000207 RTS PC ;;RETURN
5136
5137 ;MESSAGE TABLE*****
5138
5139 020364 022445 046524 031460 MSG1: .ASCII /%TM03-TU45 CONTROL LOGIC TEST- PART I (CZTUOB0)/
020372 052055 032125 020065
020400 047503 052116 047522
020406 020114 047514 044507
020414 020103 042524 052123
020422 020055 040520 052122
020430 044440 024040 055103
020436 052524 041117 024460
5140 020444 025045 025052 051501 .ASCII /%***ASSURE TAPE IS AT BOT***/
020452 052523 042522 052040
020460 050101 020105 051511
020466 040440 020124 047502
020474 025124 025052
5141 020500 052045 050131 020105 .ASCII /%TYPE <CR> TO TERMINATE RESPONSE & ^C TO RESTART%/

```

	020506	041474	037122	052040		
	020514	020117	042524	046522		
	020522	047111	052101	020105		
	020530	042522	050123	047117		
	020536	042523	023040	057040		
	020544	020103	047524	051040		
	020552	051505	040524	052122		
	020560	021445				
5142	020562	042045	044522	042526	MSG2:	.ASCII /%DRIVE NUMBER OR (CR) WHEN DONE #/
	020570	047040	046525	042502		
	020576	020122	051117	024040		
	020604	051103	020051	044127		
	020612	047105	042040	047117		
	020620	020105	043			
5143	020623	045	043045	051117	MSG2A:	.ASCII /%%FOR DRIVE ADDRESS TEST; /
	020630	042040	044522	042526		
	020636	040440	042104	042522		
	020644	051523	052040	051505		
	020652	035524				
5144	020654	020045	047105	042524		.ASCII /% ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
	020662	020122	054105	052120		
	020670	042040	044522	042526		
	020676	047040	046525	042502		
	020704	026122	040440	046114		
	020712	047440	044124	051105		
	020720	020123	044123	052517		
	020726	042114	041040	020105		
	020734	047516	026516	054105		
	020742	051511	040524	052116		
	020750	021456				
5145	020752	047045	047117	042455	MSG3:	.ASCII /%NON-EXIST DRIVE #/
	020760	044530	052123	042040		
	020766	044522	042526	021440		
5146	020774	051045	020110	042504	MSG4:	.ASCII /%RH DETECTED #/
	021002	042524	052103	042105		
	021010	021440				
5147	021012	052045	030115	020063	MSG5:	.ASCII /%TM03 DETECTED #/
	021020	042504	042524	052103		
	021026	042105	021440			
5148	021032	054105	052120	047055	MSG6:	.ASCII /%EXPT-NOT RECVD#/
	021040	052117	051040	041505		
	021046	042126	043			
5149	021051	122	053103	026504	MSG7:	.ASCII /%RCVD-NOT EXPT#/
	021056	047516	020124	054105		
	021064	052120	043			
5150	021067	045	046123	053101	MSG8:	.ASCII /%SLAVE NUMBER OR (CR) WHEN DONE #/
	021074	020105	052516	041115		
	021102	051105	047440	020122		
	021110	041450	024522	053440		
	021116	042510	020116	047504		
	021124	042516	021440			
5151	021130	022445	047506	020122	MSG8A:	.ASCII /%%FOR SLAVE ADDRESS TEST; /
	021136	046123	053101	020105		
	021144	042101	051104	051505		
	021152	020123	042524	052123		
	021160	073				

5152	021161	045	042440	052116		.ASCII	/% ENTER EXPT SLAVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
	021166	051105	042440	050130			
	021174	020124	046123	053101			
	021202	020105	052516	041115			
	021210	051105	020054	046101			
	021216	020114	052117	042510			
	021224	051522	051440	047510			
	021232	046125	020104	042502			
	021240	047040	047117	042455			
	021246	044530	052123	047101			
	021254	027124	043				
5153	021257	045	047516	026516	MSG9:	.ASCII	/%NON-EXIST SLAVE #/
	021264	054105	051511	020124			
	021272	046123	053101	020105			
	021300	043					
5154	021301	045	042522	042101	MSG10:	.ASCII	/%READ CONT BUS PAR #/
	021306	041440	047117	020124			
	021314	052502	020123	040520			
	021322	020122	043				
5155	021325	045	051127	052111	MSG11:	.ASCII	/%WRITE CONT BUS PAR #/
	021332	020105	047503	052116			
	021340	041040	051525	050040			
	021346	051101	021440				
5156	021352	042440	050130	020124	MSG12:	.ASCII	/ EXPT #/
	021360	043					
5157	021361	040	041522	042126	MSG13:	.ASCII	/ RCVD #/
	021366	021440					
5158	021370	046445	020122	044502	MSG14:	.ASCII	/%MR BITS 4-0#/
	021376	051524	032040	030055			
	021404	043					
5159	021405	045	051115	041040	MSG15:	.ASCII	/%MR BITS 15-7#/
	021412	052111	020123	032461			
	021420	033455	043				
5160	021423	045	052111	051105	MSG16:	.ASCII	/%ITER: #/
	021430	020072	043				
5161	021433	045	041524	041040	MSG18:	.ASCII	/%TC BITS 12-0 #/
	021440	052111	020123	031061			
	021446	030055	021440				
5162	021452	043045	020103	044502	MSG19:	.ASCII	/%FC BITS 15-0 #/
	021460	051524	030440	026465			
	021466	020060	043				
5163	021471	045	052506	020116	MSG20:	.ASCII	/%FUN CODE BITS 5-1 OF C1 #/
	021476	047503	042504	041040			
	021504	052111	020123	026465			
	021512	020061	043117	041440			
	021520	020061	043				
5164	021523	045	047507	041040	MSG21:	.ASCII	/%GO BIT NOT CORRECT AT START #/
	021530	052111	047040	052117			
	021536	041440	051117	042522			
	021544	052103	040440	020124			
	021552	052123	051101	020124			
	021560	043					
5165	021561	045	047507	041040	MSG22:	.ASCII	/%GO BIT NOT SET #/
	021566	052111	047040	052117			
	021574	051440	052105	021440			
5166	021602	043445	020117	044502	MSG23:	.ASCII	/%GO BIT NOT RESET BY INIT #/

	021610	020124	047516	020124		
	021616	042522	042523	020124		
	021624	054502	044440	044516		
	021632	020124	043			
5167	021635	045	051104	020131	MSG24:	.ASCII /%DRY NOT SET BY INIT #/
	021642	047516	020124	042523		
	021650	020124	054502	044440		
	021656	044516	020124	043		
5168	021663	045	051104	020131	MSG25:	.ASCII /%DRY NOT RESET BY GO=1#/
	021670	047516	020124	042522		
	021676	042523	020124	054502		
	021704	043440	036517	021461		
5169	021712	042045	054522	047040	MSG25A:	.ASCII /%DRY NOT SET BY GO=0#/
	021720	052117	051440	052105		
	021726	041040	020131	047507		
	021734	030075	043			
5170	021737	045	047516	044440	MSG26:	.ASCII /%NO INTERRUPT RETURNED#/
	021744	052116	051105	052522		
	021752	052129	051040	052105		
	021760	051125	042516	021504		
5171	021766	041045	042101	051440	MSG27:	.ASCII /%BAD STATUS#/
	021774	040524	052524	021523		
5172	022002	051445	035116	021440	MSG30:	.ASCII /%SN: #/
5173	022010	042445	051122	047040	MSG31:	.ASCII /%ERR NOT SET #/
	022016	052117	051440	052105		
	022024	021440				
5174	022026	040445	040524	047040	MSG32:	.ASCII /%ATA NOT SET #/
	022034	052117	051440	052105		
	022042	021440				
5175	022044	040445	020123	044502	MSG33:	.ASCII /%AS BIT NOT SET #/
	022052	020124	047516	020124		
	022060	042523	020124	043		
5176	022065	045	041523	047040	MSG34:	.ASCII /%SC NOT SET #/
	022072	052117	051440	052105		
	022100	021440				
5177	022102	052045	042522	047040	MSG35:	.ASCII /%TRE NOT SET #/
	022110	052117	051440	052105		
	022116	021440				
5178	022120	051445	040514	047040	MSG36:	.ASCII /%SLA NOT SET #/
	022126	052117	051440	052105		
	022134	021440				
5179	022136	051445	041523	047040	MSG37:	.ASCII /%SSC NOT SET #/
	022144	052117	051440	052105		
	022152	021440				
5180	022154	037440	021440		MSG40:	.ASCII / ? #/
5181	022160	022445	047105	020104	MSG41:	.ASCII /%END OF PASS #/
	022166	043117	050040	051501		
	022174	020123	043			
5182	022177	045	042504	042101	MSG42:	.ASCII /%DEAD TRACK #/
	022204	052040	040522	045503		
	022212	021440				
5183	022214	022445	040515	052516	MSG43:	.ASCII /%MANUAL TESTS (14-17) INHIBITED: HALT%/
	022222	046101	052040	051505		
	022230	051524	024040	032061		
	022236	030455	024467	044440		
	022244	044116	041111	052111		

	022734	046123	053101	035105	
	022742	021440			
5198	022744	021445			MSG59: .ASCII /X#/
5199	022746	021534			MSG60: .ASCII /X#/
5200	022750	051045	046505	053117	MSG62: .ASCII /XREMOVE TMDP FROM SLAVE TO BE TESTEDX#/
	022756	020105	046524	050104	
	022764	043040	047522	020115	
	022772	046123	053101	020105	
	023000	047524	041040	020105	
	023006	042524	052123	042105	
	023014	021445			
5201	023016	044045	051101	053504	MSG63: .ASCII /XHARDWARE SWR IN USEX#/
	023024	051101	020105	053523	
	023032	020122	047111	052440	
	023040	042523	021445		


```

5203                                     ;TEST HEADER*****
5204
5205 023044 022445 047514 044507 MSLT1: .ASCII /%%LOGIC TEST 1: DRIVE ADDRESSING (M8939 RH)#/
      023052 020103 042524 052123
      023060 030440 020072 051104
      023066 053111 020105 042101
      023074 051104 051505 044523
      023102 043516 024040 034115
      023110 031471 020071 044122
      023116 021451
5206 023120 022445 047514 044507 MSLT2: .ASCII /%%LOGIC TEST 2: REGISTER ADDRESSING (M8939 RH)#/
      023126 020103 042524 052123
      023134 031040 020072 042522
      023142 044507 052123 051105
      023150 040440 042104 042522
      023156 051523 047111 020107
      023164 046450 034470 034463
      023172 051040 024510 043
5207 023177 045 046045 043517 MSLT3: .ASCII /%%LOGIC TEST 3: CONTROL BUS TEST (RH M8905-YB M8939)#/
      023204 041511 052040 051505
      023212 020124 035063 041440
      023220 047117 051124 046117
      023226 041040 051525 052040
      023234 051505 020124 051050
      023242 020110 034115 030071
      023250 026465 041131 046440
      023256 034470 034463 021451
5208 023264 022445 047514 044507 MSLT4: .ASCII /%%LOGIC TEST 4: SLAVE ADDRESSING (M8905-YB M8933)#/
      023272 020103 042524 052123
      023300 032040 020072 046123
      023306 053101 020105 042101
      023314 051104 051505 044523
      023322 043516 024040 034115
      023330 030071 026465 041131
      023336 046440 034470 031463
      023344 021451
5209 023346 022445 047514 044507 MSLT5: .ASCII /%%LOGIC TEST 5: MR BIT TEST (M8905-YB)#/
      023354 020103 042524 052123
      023362 032440 020072 051115
      023370 041040 052111 052040
      023376 051505 020124 046450
      023404 034470 032460 054455
      023412 024502 043
5210 023415 045 046045 043517 MSLT6: .ASCII /%%LOGIC TEST 6: TC BIT TEST (M8905-YB)#/
      023422 041511 052040 051505
      023430 020124 035066 052040
      023436 020103 044502 020124
      023444 042524 052123 024040
      023452 034115 030071 026465
      023460 041131 021451
5211 023464 022445 047514 044507 MSLT7: .ASCII /%%LOGIC TEST 7: FC BIT TEST (M8905-YB)#/
      023472 020103 042524 052123
      023500 033440 020072 041506
      023506 041040 052111 052040
      023514 051505 020124 046450
      023522 034470 032460 054455
  
```

	023530	024502	043		
5212	023533	045	046045	043517	MSLT10: .ASCII /%%LOGIC TEST 10: FUNCTION BIT TEST (M8905-YB)#/
	023540	041511	052040	051505	
	023546	020124	030061	020072	
	023554	052506	041516	044524	
	023562	047117	041040	052111	
	023570	052040	051505	020124	
	023576	046450	034470	032460	
	023604	054455	024502	043	
5213	023611	045	046045	043517	MSLT11: .ASCII /%%LOGIC TEST 11: GO BIT TEST (M8939)#/
	023616	041511	052040	051505	
	023624	020124	030461	020072	
	023632	047507	041040	052111	
	023640	052040	051505	020124	
	023646	046450	034470	034463	
	023654	021451			
5214	023656	022445	047514	044507	MSLT12: .ASCII /%%LOGIC TEST 12: DRIVE READY BIT (M8939)#/
	023664	020103	042524	052123	
	023672	030440	035062	042040	
	023700	044522	042526	051040	
	023706	040505	054504	041040	
	023714	052111	024040	034115	
	023722	031471	024471	043	
5215	023727	045	046045	043517	MSLT13: .ASCII /%%LOGIC TEST 13: INTERRUPT TEST (RH)#/
	023734	041511	052040	051505	
	023742	020124	031461	020072	
	023750	047111	042524	051122	
	023756	050125	020124	042524	
	023764	052123	024040	044122	
	023772	021451			
5216	023774	022445	047514	044507	MSLT14: .ASCII /%%LOGIC TEST 14: MANUAL STATUS TEST 1#/
	024002	020103	042524	052123	
	024010	030440	035064	046440	
	024016	047101	040525	020114	
	024024	052123	052101	051525	
	024032	052040	051505	020124	
	024040	021461			
5217	024042	022445	047514	044507	MSLT15: .ASCII /%%LOGIC TEST 15: MANUAL STATUS TEST 2#/
	024050	020103	042524	052123	
	024056	030440	035065	046440	
	024064	047101	040525	020114	
	024072	052123	052101	051525	
	024100	052040	051505	020124	
	024106	021462			
5218	024110	022445	047514	044507	MSLT16: .ASCII /%%LOGIC TEST 16: MANUAL STATUS TEST 3#/
	024116	020103	042524	052123	
	024124	030440	035066	046440	
	024132	047101	040525	020114	
	024140	052123	052101	051525	
	024146	052040	051505	020124	
	024154	021463			
5219	024156	022445	047514	044507	MSLT17: .ASCII /%%LOGIC TEST 17: MANUAL STATUS TEST 4#/
	024164	020103	042524	052123	
	024172	030440	035067	046440	
	024200	047101	040525	020114	
	024206	052123	052101	051525	

	024214	052040	051505	020124	
	024222	021464			
5220	024224	022445	047514	044507	MSLT20: .ASCII /%%LOGIC TEST 20: ILLEGAL FUNCTION TEST (M8939)#/
	024232	020103	042524	052123	
	024240	031040	035060	044440	
	024246	046114	043505	046101	
	024254	043040	047125	052103	
	024262	047511	020116	042524	
	024270	052123	024040	034115	
	024276	031471	024471	043	
5221	024303	045	046045	043517	MSLT21: .ASCII /%%LOGIC TEST 21: RMR(M8939)#/
	024310	041511	052040	051505	
	024316	020124	030462	020072	
	024324	046522	024122	034115	
	024332	031471	024471	043	
5222	024337	045	046045	043517	MSLT22: .ASCII /%%LOGIC TEST 22: CPAR(M8939)#/
	024344	041511	052040	051505	
	024352	020124	031062	020072	
	024360	050103	051101	046450	
	024366	034470	034463	021451	
5223	024374	022445	047514	044507	MSLT23: .ASCII /%%LOGIC TEST 23: FMT(M8905-YB M8906)#/
	024402	020103	042524	052123	
	024410	031040	035063	043040	
	024416	052115	046450	034470	
	024424	032460	054455	020102	
	024432	034115	030071	024466	
	024440	043			
5224	024441	045	046045	043517	MSLT24: .ASCII /%%LOGIC TEST 24: DPAR(M8906 RH)#/
	024446	041511	052040	051505	
	024454	020124	032062	020072	
	024462	050104	051101	046450	
	024470	034470	033060	051040	
	024476	024510	043		
5225	024501	045	046045	043517	MSLT25: .ASCII /%%LOGIC TEST 25: NEF(M8939)#/
	024506	041511	052040	051505	
	024514	020124	032462	020072	
	024522	042516	024106	034115	
	024530	031471	024471	043	
5226	024535	045	046045	043517	MSLT26: .ASCII /%%LOGIC TEST 26: FCE(M8939)#/
	024542	041511	052040	051505	
	024550	020124	033062	020072	
	024556	041506	024105	034115	
	024564	031471	024471	043	
5227	024571	045	046045	043517	MSLT27: .ASCII /%%LOGIC TEST 27: ILR(M8939)#/
	024576	041511	052040	051505	
	024604	020124	033462	020072	
	024612	046111	024122	034115	
	024620	031471	024471	043	
5228	024625	045	046045	043517	MSLT30: .ASCII /%%LOGIC TEST 30:DTE(M8906 RH)#/
	024632	041511	052040	051505	
	024640	020124	030063	042072	
	024646	042524	046450	034470	
	024654	033060	051040	024510	
	024662	043			
5229	024663	045	046045	043517	MSLT31: .ASCII /%%LOGIC TEST 31: OPI(M8933)#/
	024670	041511	052040	051505	

	024676	020124	030463	020072	
	024704	050117	024111	034115	
	024712	031471	024463	043	
5230	024717	045	046045	043517	MSLT32: .ASCII /%%LOGIC TEST 32: UNS(M8939)#/
	024724	041511	052040	051505	
	024732	020124	031063	020072	
	024740	047125	024123	034115	
	024746	031471	024471	043	
5231	024753	045	046045	043517	MSLT33: .ASCII /%%LOGIC TEST 33: PIP(M8939)#/
	024760	041511	052040	051505	
	024766	020124	031463	020072	
	024774	044520	024120	034115	
5232	025002	031471	024471	043	MSLT34: .ASCII /%%LOGIC TEST 34: PES(M8928)#/
	025007	045	046045	043517	
	025014	041511	052040	051505	
	025022	020124	032063	020072	
	025030	042520	024123	034115	
5233	025036	031071	024470	043	MSLT35: .ASCII /%%LOGIC TEST 35: SAC(M8933 M8905-YB)#/
	025043	045	046045	043517	
	025050	041511	052040	051505	
	025056	020124	032463	020072	
	025064	040523	024103	034115	
	025072	031471	020063	034115	
	025100	030071	026465	041131	
	025106	021451			
5234	025110	022445	047514	044507	MSLT36: .ASCII /%%LOGIC TEST 36: FCS(M8933 M8905-YB)#/
	025116	020103	042524	052123	
	025124	031440	035066	043040	
	025132	051503	046450	034470	
	025140	031463	046440	034470	
	025146	032460	054455	024502	
	025154	043			
5235	025155	045	046045	043517	MSLT37: .ASCII /%%LOGIC TEST 37: ACCL(M8933 M8905-YB)#/
	025162	041511	052040	051505	
	025170	020124	033463	020072	
	025176	041501	046103	046450	
	025204	034470	031463	046440	
	025212	034470	032460	054455	
	025220	024502	043		
5236	025223	045	046045	043517	MSLT40: .ASCII /%%LOGIC TEST 40: PE TAPE MARK(M8932)#/
	025230	041511	052040	051505	
	025236	020124	030064	020072	
	025244	042520	052040	050101	
	025252	020105	040515	045522	
	025260	046450	034470	031063	
	025266	021451			
5237	025270	022445	047514	044507	MSLT41: .ASCII /%%LOGIC TEST 41: NRZ TAPE MARK (M8934)#/
	025276	020103	042524	052123	
	025304	032040	035061	047040	
	025312	055122	052040	050101	
	025320	020105	040515	045522	
	025326	024040	034115	031471	
	025334	024464	043		
5238	025337	045	046045	043517	MSLT42: .ASCII /%%LOGIC TEST 42: CRC(M8934)#/
	025344	041511	052040	051505	
	025352	020124	031064	020072	

	025360	051103	024103	034115	
	025366	031471	024464	043	
5239	025373	045	046045	043517	MSLT43: .ASCII /%%LOGIC TEST 43: LRC(M8934)#/
	025400	041511	052040	051505	
	025406	020124	031464	020072	
	025414	051114	024103	034115	
	025422	031471	024464	043	
5240	025427	045	046045	043517	MSLT44: .ASCII /%%LOGIC TEST 44: CORRECTABLE DATA (M8932 M8901)#/
	025434	041511	052040	051505	
	025442	020124	032064	020072	
	025450	047503	051122	041505	
	025456	040524	046102	020105	
	025464	040504	040524	024040	
	025472	034115	031471	020062	
	025500	034115	030071	024461	
	025506	043			
5241	025507	045	046045	043517	MSLT45: .ASCII /%%LOGIC TEST 45: INCORRECTABLE DATA (M8932 M8934)#/
	025514	041511	052040	051505	
	025522	020124	032464	020072	
	025530	047111	047503	051122	
	025536	041505	040524	046102	
	025544	020105	040504	040524	
	025552	024040	034115	031471	
	025560	020062	034115	031471	
	025566	024464	043		
5242	025571	045	046045	043517	MSLT46: .ASCII /%%LOGIC TEST 46: PEF ('93?)#/
	025576	041511	052040	051505	
	025604	020124	033064	020072	
	025612	042520	024106	034115	
	025620	031471	024464	043	
5243	025625	045	046045	043517	MSLT47: .ASCII /%%LOGIC TEST 47: FC OVERFLOW (M8905-YB)#/
	025632	041511	052040	051505	
	025640	020124	033464	020072	
	025646	041506	047440	042526	
	025654	043122	047514	020127	
	025662	046450	034470	032460	
	025670	054455	024502	043	
5244	025675	045	046045	043517	MSLT50: .ASCII /%%LOGIC TEST 50: NEF WHEN WRITE PE ON NRZ SLAVE#/
	025702	041511	052040	051505	
	025710	020124	030065	020072	
	025716	042516	020106	044127	
	025724	047105	053440	044522	
	025732	042524	050040	020105	
	025740	047117	047040	055122	
	025746	051440	040514	042526	
	025754	043			
5245	025755	045	046045	043517	MSLT51: .ASCII /%%LOGIC TEST 51: NEF WHEN WRITE NRZ ON PE SLAVE#/
	025762	041511	052040	051505	
	025770	020124	030465	020072	
	025776	042516	020106	044127	
	026004	047105	053440	044522	
	026012	042524	047040	055122	
	026020	047440	020116	042520	
	026026	051440	040514	042526	
	026034	043			

```

5247
5248                                     :MANUAL INSTRUCTION*****
5249
5250 026035      045  054524  042520 MMSG0: .ASCII  /%TYPE CR WHEN READY;#/
      026042  041440 020122  044127
      026050  047105 051040  040505
      026056  054504 021473
5251 026062  022445 047515  047125 MMSG1: .ASCII  /%%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE:#/
      026070  020124 040524  042520
      026076  053440 052111  020110
      026104  047516 053440  044522
      026112  042524 051040  047111
      026120  026107 046040  040517
      026126  020104 047524  041040
      026134  052117 020054  042523
      026142  020124 047524  047440
      026150  020116 044514  042516
      026156  021472
5252 026160  051445 052105  052040 MMSG2: .ASCII  /%SET TO OFFLINE:#/
      026166  020117 043117  046106
      026174  047111 035105   043
5253 026201      045 047515  042526 MMSG3: .ASCII  /%MOVE FORWARD TO EOT, USING THE TU45 MAINTENENCE SWITCH, SET ONLINE.#/
      026206  043040 051117  040527
      026214  042122 052040  020117
      026222  047505 026124  052440
      026230  044523 043516  052040
      026236  042510 052040  032125
      026244  020065 040515  047111
      026252  042524 042516  041516
      026260  020105 053523  052111
      026266  044103 020054  042523
      026274  020124 047117  044514
      026302  042516 021472
5254 026306  051045 053505  047111 MMSG4: .ASCII  /%REWIND AND UNLOAD TAPE, REINSTALL WRITE RING, AND/
      026314  020104 047101  020104
      026322  047125 047514  042101
      026330  052040 050101  026105
      026336  051040 044505  051516
      026344  040524 046114  053440
      026352  044522 042524  051040
      026360  047111 026107  040440
      026366  042116
5255 026370  046445 053117  020105 .ASCII  /%MOVE JUST FORWARD OF BOT USING THE TU45 MAINTENENCE SWITCH,/
      026376  052512 052123  043040
      026404  051117 040527  042122
      026412  047440 020106  047502
      026420  020124 051525  047111
      026426  020107 044124  020105
      026434  052524 032464  046440
      026442  044501 052116  047105
      026450  047105 042503  051440
      026456  044527 041524  026110
5256 026464  040445 042116  051440 .ASCII  /%AND SET TO ONLINE#/
      026472  052105 052040  020117
      026500  047117 044514  042516
      026506      043
  
```

5257	026507	045	046445	053117	MMSG5:	.ASCII	/%MOVE TAPE TO BOT; ON LINE#/ 026514 020105 040524 042520 026522 052040 020117 047502 026530 035524 047440 020116 026536 044514 042516 043
5258	026543	045	020045	051104	MMSG6:	.ASCII	/% DRIVE #/ 026550 053111 020105 043
5259	026555	054	046123	053101	MMSG7:	.ASCII	/.SLAVE #/ 026562 020105 043
5260	026565	116	052117	047440	MMSG8:	.ASCII	/NOT ON LINE # / 026572 020116 044514 042516 026600 021440 040

```

5262
5263           ;TAG MESSAGE
5264
5265 026603      045 053523 020122 SMSWR: .ASCII /%SWR = #/
      026610      020075      043
5266 026613      040 042516 020127 SMNEW: .ASCII / NEW = #/
      026620      020075      043
5267 026623      045 046123 020101 TMS1: .ASCII /%SLA #/
      026630      043
5268 026631      045 047502 020124 TMS2: .ASCII /%BOT #/
      026636      043
5269 026637      045 046524 021440 TMS3: .ASCII /%TM #/
5270 026644      044445 041104 021440 TMS4: .ASCII /%IDB #/
5271 026652      051445 053504 020116 TMS5: .ASCII /%SDWN #/
      026660      043
5272 026661      045 042520 020123 TMS6: .ASCII /%PES #/
      026666      043
5273 026667      045 051523 020103 TMS7: .ASCII /%SSC #/
      026674      043
5274 026675      045 051104 020131 TMS8: .ASCII /%DRY #/
      026702      043
5275 026703      045 050104 020122 TMS9: .ASCII /%DPR #/
      026710      043
5276 026711      045 052116 020114 TMS10: .ASCII /%NTL #/
      026716      043
5277 026717      045 047505 020124 TMS11: .ASCII /%EOT #/
      026724      043
5278 026725      045 051127 020114 TMS12: .ASCII /%WRL #/
      026732      043
5279 026733      045 047515 020114 TMS13: .ASCII /%MOL #/
      026740      043
5280 026741      045 044520 020120 TMS14: .ASCII /%PIP #/
      026746      043
5281 026747      045 051105 020122 TMS15: .ASCII /%ERR #/
      026754      043
5282 026755      045 052101 020101 TMS16: .ASCII /%ATA #/
      026762      043
5283 026763      045 046111 020106 TMS17: .ASCII /%ILF #/
      026770      043
5284 026771      045 046111 020122 TMS18: .ASCII /%ILR #/
      026776      043
5285 026777      045 046522 020122 TMS19: .ASCII /%RMR #/
      027004      043
5286 027005      045 050103 051101 TMS20: .ASCII /%CPAR #/
      027012      021440
5287 027014      043045 052115 021440 TMS21: .ASCII /%FMT #/
5288 027022      042045 040520 020122 TMS22: .ASCII /%DPAR #/
      027030      043
5289 027031      045 047111 020103 TMS23: .ASCII /%INC #/
      027036      043
5290 027037      045 050126 020105 TMS24: .ASCII /%VPE #/
      027044      043
5291 027045      045 042520 020106 TMS25: .ASCII /%PEF #/
      027052      043
5292 027053      045 051114 020103 TMS26: .ASCII /%LRC #/
      027060      043
  
```


5293	027061	045	051516	020107	TMS27:	.ASCII	/XMSG #/
	027066	043					
5294	027067	045	041506	020105	TMS28:	.ASCII	/XFCF #/
	027074	043					
5295	027075	045	051503	021440	TMS29:	.ASCII	/XCS #/
5296	027102	044445	046524	021440	TMS30:	.ASCII	/XITM #/
5297	027110	047045	043105	021440	TMS31:	.ASCII	/XNEF #/
5298	027116	042045	042524	021440	TMS32:	.ASCII	/XDTE #/
5299	027124	047445	044520	021440	TMS33:	.ASCII	/XOPI #/
5300	027132	053445	044522	042524	TMS33A:	.ASCII	/XWRITE OPI #/
	027140	047440	044520	021440			
5301	027146	051045	040505	020104	TMS33B:	.ASCII	/XREAD OPI #/
	027154	050117	020111	043			
5302	027161	040	041517	052503	TMS33C:	.ASCII	/ OCCURED TO SOONX#/
	027166	042522	020104	047524			
	027174	051440	047517	022516			
	027202	043					
5303	027203	040	041517	052503	TMS33D:	.ASCII	/ OCCURRED TO LATEX#/
	027210	051122	042105	052040			
	027216	020117	040514	042524			
	027224	021445					
5304	027226	043040	044501	042514	TMS33E:	.ASCII	/ FAILED TO SETX#/
	027234	020104	047524	051440			
	027242	052105	021445				
5305	027246	052445	051516	021440	TMS34:	.ASCII	/XUNS #/
5306	027254	041445	051117	020122	TMS35:	.ASCII	/XCORR #/
	027262	043					
5307	027263	045	051103	020103	TMS36:	.ASCII	/XCRC #/
	027270	043					
5308	027271	045	040523	020103	TMS37:	.ASCII	/XSAC #/
	027276	043					
5309	027277	045	041506	020123	TMS38:	.ASCII	/XFCS #/
	027304	043					
5310	027305	045	041501	046103	TMS39:	.ASCII	/XACCL #/
	027312	021440					

5311							
5312						.EVEN	
5313						:WRITE	BUFFER
5314							
5315	027314	000100			WDATA:		
5317	027314	177777				-1	
(1)	027316	177777				-1	
(1)	027320	177777				-1	
(1)	027322	177777				-1	
(1)	027324	177777				-1	
(1)	027326	177777				-1	
(1)	027330	177777				-1	
(1)	027332	177777				-1	
(1)	027334	177777				-1	
(1)	027336	177777				-1	
(1)	027340	177777				-1	
(1)	027342	177777				-1	
(1)	027344	177777				-1	
(1)	027346	177777				-1	
(1)	027350	177777				-1	
(1)	027352	177777				-1	

(1)	027354	177777	-1
(1)	027356	177777	-1
(1)	027360	177777	-1
(1)	027362	177777	-1
(1)	027364	177777	-1
(1)	027366	177777	-1
(1)	027370	177777	-1
(1)	027372	177777	-1
(1)	027374	177777	-1
(1)	027376	177777	-1
(1)	027400	177777	-1
(1)	027402	177777	-1
(1)	027404	177777	-1
(1)	027406	177777	-1
(1)	027410	177777	-1
(1)	027412	177777	-1
(1)	027414	177777	-1
(1)	027416	177777	-1
(1)	027420	177777	-1
(1)	027422	177777	-1
(1)	027424	177777	-1
(1)	027426	177777	-1
(1)	027430	177777	-1
(1)	027432	177777	-1
(1)	027434	177777	-1
(1)	027436	177777	-1
(1)	027440	177777	-1
(1)	027442	177777	-1
(1)	027444	177777	-1
(1)	027446	177777	-1
(1)	027450	177777	-1
(1)	027452	177777	-1
(1)	027454	177777	-1
(1)	027456	177777	-1
(1)	027460	177777	-1
(1)	027462	177777	-1
(1)	027464	177777	-1
(1)	027466	177777	-1
(1)	027470	177777	-1
(1)	027472	177777	-1
(1)	027474	177777	-1
(1)	027476	177777	-1
(1)	027500	177777	-1
(1)	027502	177777	-1
(1)	027504	177777	-1
(1)	027506	177777	-1
(1)	027510	177777	-1
(1)	027512	177777	-1

5318
5319
5320
5321

:READ BUFFER

5322	027514	000100
5324	027514	000000
(1)	027516	000000
(1)	027520	000000

RDATA:
0
0
0

(1)	027522	000000	0
(1)	027524	000000	0
(1)	027526	000000	0
(1)	027530	000000	0
(1)	027532	000000	0
(1)	027534	000000	0
(1)	027536	000000	0
(1)	027540	000000	0
(1)	027542	000000	0
(1)	027544	000000	0
(1)	027546	000000	0
(1)	027550	000000	0
(1)	027552	000000	0
(1)	027554	000000	0
(1)	027556	000000	0
(1)	027560	000000	0
(1)	027562	000000	0
(1)	027564	000000	0
(1)	027566	000000	0
(1)	027570	000000	0
(1)	027572	000000	0
(1)	027574	000000	0
(1)	027576	000000	0
(1)	027600	000000	0
(1)	027602	000000	0
(1)	027604	000000	0
(1)	027606	000000	0
(1)	027610	000000	0
(1)	027612	000000	0
(1)	027614	000000	0
(1)	027616	000000	0
(1)	027620	000000	0
(1)	027622	000000	0
(1)	027624	000000	0
(1)	027626	000000	0
(1)	027630	000000	0
(1)	027632	000000	0
(1)	027634	000000	0
(1)	027636	000000	0
(1)	027640	000000	0
(1)	027642	000000	0
(1)	027644	000000	0
(1)	027646	000000	0
(1)	027650	000000	0
(1)	027652	000000	0
(1)	027654	000000	0
(1)	027656	000000	0
(1)	027660	000000	0
(1)	027662	000000	0
(1)	027664	000000	0
(1)	027666	000000	0
(1)	027670	000000	0
(1)	027672	000000	0
(1)	027674	000000	0
(1)	027676	000000	0
(1)	027700	000000	0

(1) 027702 000000 0
(1) 027704 000000 0
(1) 027706 000000 0
(1) 027710 000000 0
(1) 027712 000000 0

5325
5326
5327

;WRAP AROUND MESSAGES*****

5328 027714 042045 052101 020101 WMSG27: .ASCII /%DATA PAT:#/

027722 040520 035124 043

5329 027727 045 047505 020122 WMSG31: .ASCII /%EOR CLEAR DID NOT CLEAR GO%#/

027734 046103 040505 020122

027742 044504 020104 047516

027750 020124 046103 040505

027756 020122 047507 021445

5330
5331

PRE: .EVEN

5332 027764 000000 0

5335 027766 000000 0

(1) 027770 000000 0

(1) 027772 000000 0

(1) 027774 000000 0

(1) 027776 000000 0

(1) 030000 000000 0

(1) 030002 000000 0

(1) 030004 000000 0

(1) 030006 000000 0

(1) 030010 000000 0

(1) 030012 000000 0

(1) 030014 000000 0

(1) 030016 000000 0

(1) 030020 000000 0

(1) 030022 000000 0

(1) 030024 000000 0

(1) 030026 000000 0

(1) 030030 000000 0

(1) 030032 000000 0

(1) 030034 000000 0

(1) 030036 000000 0

(1) 030040 000000 0

(1) 030042 000000 0

(1) 030044 000000 0

(1) 030046 000000 0

(1) 030050 000000 0

(1) 030052 000000 0

(1) 030054 000000 0

(1) 030056 000000 0

(1) 030060 000000 0

(1) 030062 000000 0

(1) 030064 000000 0

(1) 030066 000000 0

(1) 030070 000000 0

(1) 030072 000000 0

(1) 030074 000000 0

(1) 030076 000000 0

(1) 030100 000000 0

(1)	030102	000000		0
(1)	030104	000000		0
5336	030106	000000	POST:	0
5339	030110	000000		0
(1)	030112	000000		0
(1)	030114	000000		0
(1)	030116	000000		0
(1)	030120	000000		0
(1)	030122	000000		0
(1)	030124	000000		0
(1)	030126	000000		0
(1)	030130	000000		0
(1)	030132	000000		0
(1)	030134	000000		0
(1)	030136	000000		0
(1)	030140	000000		0
(1)	030142	000000		0
(1)	030144	000000		0
(1)	030146	000000		0
(1)	030150	000000		0
(1)	030152	000000		0
(1)	030154	000000		0
(1)	030156	000000		0
(1)	030160	000000		0
(1)	030162	000000		0
(1)	030164	000000		0
(1)	030166	000000		0
(1)	030170	000000		0
(1)	030172	000000		0
(1)	030174	000000		0
(1)	030176	000000		0
(1)	030200	000000		0
(1)	030202	000000		0
(1)	030204	000000		0
(1)	030206	000000		0
(1)	030210	000000		0
(1)	030212	000000		0
(1)	030214	000000		0
(1)	030216	000000		0
(1)	030220	000000		0
(1)	030222	000000		0
(1)	030224	000000		0
(1)	030226	000000		0
5340	030230	000000	WBUF:	0
5341		030642		=.+410
5342	030642	000000	RBUF:	0
5343				
5344		000001		.END

LT15A	005432	3417	3421#	
LT15IT	005450	2780	3424#	3430
LT15X	005514	3429	3433#	
LT15XX	005520	3419	3434#	
LT16	005524	2781	3439#	
LT16A	005546	3440	3444#	
LT16IT	005564	2782	3447#	3453
LT16X	005630	3452	3456#	
LT16XX	005634	3442	3457#	
LT17	005640	2783	3463#	
LT17A	005662	3464	3468#	
LT17IT	005700	2784	3471#	3477
LT17X	005744	3476	3480#	
LT17XX	005750	3466	3481#	
LT2	003120	2757	2758	3051#
LT2A	003160	3058#	3068	3082
LT2B	003200	3061	3063#	
LT2C	003214	3064	3066#	
LT2ERG	003274	3072	3075	3078#
LT2ER1	003224	3062	3070#	
LT2ER2	003242	3065	3073#	
LT2ER3	003260	3076#		
LT2IT	003122	3052#		
LT2LP	003310	3078	3081#	
LT2X	003314	3069	3083#	
LT20	005754	2785	3491#	
LT20A	006002	3492	3495#	3512
LT20B	006060	3501	3506#	
LT20C	006070	3505	3507	3509#
LT20IT	005770	2786	3493#	
LT20X	006102	3510	3513#	
LT21	006116	2787	3520#	
LT21A	006222	3530	3535#	
LT21B	006232	3534	3536	3538#
LT21IT	006132	2788	3521	3522#
LT21XA	006242	3540#	3541	
LT22	006262	2789	3549#	
LT22A	006362	3558	3563#	
LT22IT	006276	2790	3550	3551#
LT22X	006372	3562	3564	3566#
LT23	006406	2791	3573#	
LT23A	006506	3582	3587#	
LT23IT	006422	2792	3574	3575#
LT23X	006516	3586	3588	3590#
LT24	006536	2793	3597#	
LT24B	006664	3616#	3622	
LT24B0	006706	3619	3621#	
LT24C	006720	3625#	3626	
LT24D	007002	3628	3639#	
LT24IT	006552	2794	3598	3599#
LT24X	007026	3638	3643	3645#
LT25	007062	2795	3655#	
LT25A	007170	3664	3670#	
LT25IT	007070	2796	3656#	3665
LT25X	007200	3669	3671	3673#
LT26	007214	2797	3680#	

LT26IT	007222	2798	3681#	3705			
LT26X	007414	3709	3712	3714#			
LT27	007430	2799	3721#				
LT27A	007466	3726	3730#	3740			
LT27B	007524	3733	3737#				
LT27IT	007456	2800	3728#				
LT27X	007534	3738	3741#				
LT27XX	007544	3722	3725	3743#			
LT3	003324	2759	3088#				
LT3A	003344	3092#	3102	3104	3116		
LT3B	003364	3096#	3107	3109			
LT3C	003400	3099#	3111				
LT3ER1	003410	3095	3103#				
LT3ER2	003436	3098	3108#				
LT3IT	003326	2760	3089#				
LT3X	003460	3100	3112#				
LT3XX	003476	3113	3117#				
LT30	007550	2801	3748#				
LT30A	007652	3758	3761#				
LT30B	007714	3768#	3771				
LT30C	007754	3769	3778#				
LT30D	007772	3781#	3782				
LT30E	010022	3784	3788#				
LT30IT	007572	2802	3750	3751#			
LT30X	010046	3777	3787	3792	3794#		
LT31	010066	2803	3803#				
LT31A	010244	3829	3831#				
LT31IT	010074	2804	3804#				
LT31X	010350	3825	3827	3848	3850#	3898	
LT32	010700	2805	3939#				
LT32IT	010714	2806	3940	3941#			
LT32X	011026	3957	3959	3961#			
LT32XX	011036	3948	3963#				
LT33	011042	2807	3973#				
LT33IT	011056	2808	3974	3975#			
LT33X	011134	3980	3984#				
LT34	011144	2809	3990#				
LT34A	011204	3996#					
LT34A1	011164	3993#	3999				
LT34B	011234	3997	4001#				
LT34C	011240	4002#	4004				
LT34IT	011160	2810	3992#				
LT34X	011270	4003	4007#				
LT34XX	011274	4008#					
LT35	011300	2811	4013#				
LT35A	011402	4022	4026#	4030			
LT35IT	011314	2812	4015#	4024			
LT35X	011442	4029	4033#				
LT36	011452	2813	4039#				
LT36IT	011466	2814	4041#	4044			
LT36X	011562	4050	4054#				
LT37	011572	2815	4060#				
LT37A	011646	4065	4069#	4078			
LT37B	011700	4074#	4077				
LT37IT	011606	2816	4062#	4067			
LT37X	011732	4075	4081#				

LT4	003506	2761	2762	3123#	
LT4A	003624	3130	3144#	3164	3170
LT4B	003666	3149	3153#		
LT4C	003674	3155#			
LT4D	003724	3151	3161#	3172	
LT4ERG	003752	3166	3168#		
LT4ER1	003734	3152	3165#		
LT4ER2	003744	3154	3167#		
LT4G	003554	3133#	3162		
LT4GO	003544	3131#			
LT4X	004002	3128	3141	3160	3173#
LT40	011742	2817	4087#		
LT40IT	011756	2818	4087	4089#	
LT40X	012052	4098	4104#		
LT40XX	012056	4105#			
LT41	012062	2819	4110#		
LT41IT	012076	2820	4110	4112#	
LT41X	012272	4135	4139	4141#	
LT42	012324	2821	4157#		
LT42A	012426	4171#	4174		
LT42B	012442	4172	4175#		
LT42B1	012470	4176	4181#		
LT42B2	012510	4180	4185#		
LT42C	012554	4192#	4195		
LT42D	012570	4193	4196#		
LT42E	012620	4197	4202#		
LT42IT	012362	2822	4163	4164#	
LT42X	012640	4201	4205	4207#	
LT43	012654	2823	4213#		
LT43C	012754	4227#	4230		
LT43D	012770	4228	4231#		
LT43E	013022	4232	4237#		
LT43F	013054	4241	4244#		
LT43IT	012702	2824	4215	4217#	
LT43X	013074	4236	4247	4249#	
LT44	013104	2825	4254#		
LT44A	013170	4264#	4267		
LT44A1	013202	4265	4268#		
LT44B	013212	4270#	4273		
LT44C	013226	4271	4274#		
LT44D	013254	4275	4279#		
LT44E	013256	4280#			
LT44F	013316	4281	4288#		
LT44IT	013132	2826	4257	4258#	
LT44X	013336	4287	4291	4293#	
LT44XX	013346	4295#			
LT45	013352	2827	4300#		
LT45A	013460	4316#	4319		
LT45B	013474	4317	4320#		
LT45D	013524	4321	4325#		
LT45E	013444	4311#	4314		
LT45E1	013456	4312	4315#		
LT45F	013556	4329	4332#		
LT45IT	013400	2828	4303	4304#	
LT45X	013602	4336	4338#		
LT45XX	013612	4340#			

LT46	013616	2829	4345#																		
LT46A	013700	4354#	4355																		
LT46B	013746	4364#	4367																		
LT46C	013762	4365	4368#																		
LT46D	014014	4369	4374#																		
LT46IT	013644	2830	4348	4349#																	
LT46X	014040	4373	4378	4380#																	
LT46XX	014050	4382#																			
LT47	014054	2831	4386#																		
LT47A	014144	4396#	4399																		
LT47B	014156	4397	4400#																		
LT47C	014162	4401#	4404																		
LT47IT	014070	2832	4387	4388#																	
LT47X	014234	4408	4412#																		
LT5	004006	2763	3178#																		
LT5A	004026	3182#	3190	3204																	
LT5B	004046	3187#	3206																		
LT5C	004056	3188	3191#																		
LT5D	004066	3193#	3202	3208																	
LT5E	004110	3197	3199#	3210																	
LT5ER1	004122	3186	3203#																		
LT5ER2	004144	3198	3207#																		
LT5IT	004014	2764	3179#																		
LT5X	004166	3200	3211#																		
LT5O	014244	2833	4419#																		
LT5OIT	014260	2834	4420	4421#																	
LT5OX	014404	4438	4440	4442#																	
LT5I	014420	2835	4447#																		
LT5IIT	014434	2836	4448	4449#																	
LT5IX	014560	4466	4468	4470#																	
LT6	004176	2765	3217#																		
LT6A	004214	3221#	3232																		
LT6A1	004212	3220#	3230																		
LT6B	004220	3222#	3234																		
LT6D	004240	3227#	3236																		
LT6ER1	004260	3226	3233#																		
LT6IT	004200	2766	3218#																		
LT6X	004302	3228	3237#																		
LT7	004312	2767	3243#																		
LT7A	004330	3247#	3257	3259																	
LT7B	004350	3252#	3261																		
LT7C	004320	3245#	3255																		
LT7ER1	004370	3251	3258#																		
LT7IT	004314	2768	3244#																		
LT7X	004412	3253	3262#																		
PPMSG0	026035	4841	5250#																		
PPMSG1	026062	3398	5251#																		
PPMSG2	026160	3422	5252#																		
PPMSG3	026201	3445	5253#																		
PPMSG4	026306	3469	5254#																		
PPMSG5	026507	5257#																			
PPMSG6	026543	2944	5258#																		
PPMSG7	026555	2948	5259#																		
PPMSG8	026565	2952	5260#																		
MR	000534	2630#	3182*	3183	3193*	3194	3302*	3342*	3524*	3578*	3605*	3616*	3620	3623							
		3659*	3688*	3694*	3695*	3754*	3765*	3778*	3779*	3812*	3836*	3862	3864	3869							

OCTP0	017620	5010	5012	5015#																
OCTP1	017640	5016	5020#																	
OCTP2	017646	5019	5022#																	
OCTP3	017736	5014	5045#																	
OFL	020014	5006*	5008*	5011	5051	5055*	5061#													
PATRN	001002	2722#																		
PCNTR	001016	2728#	2929*	2983	2995*	4763														
PEXFL	000736	2704#	4703*	4715	4741*															
PFLG	000666	2684#	4524*	4545*	4570	4613*	4647*	4651	4689*	4694	4698*	4705*	4731*	4736						
POST	030106	5336#																		
PRE	027764	5332#																		
PREFL	000770	2717#																		
PSW	000566	2649#	3374*	3378*	4865*	4967	4968*	4972*												
REBUF	030642	5342#																		
RCDP	001010	2725#																		
RDAD	000762	2714#																		
RDATA	027514	5022#																		
RDRVF	001006	2714#	2928*	4482*																
REGP	015304	4560	4580#	4646	4688															
REGS	000612	2659#	2867	2869	2886															
RTRN	000670	2685#	3372*	4854																
SAV1	000704	2691#																		
SAV2	000706	2692#																		
SAV3	000710	2693#																		
SCF	000732	2702#																		
SCOLP	000712	2694#	3043*	3078*	3104*	3109*	3170*	3204*	3208*	3234*	3259*	3287*	3317*	3323*						
		3329*	3353*	3357*	3361*	3380*	3406*	3430*	3453*	3477*	3492*	3521*	3550*	3574*						
		3598*	3665*	3705*	3726*	3750*	3804*	3829*	3940*	3974*	3999*	4004*	4024*	4030*						
		4044*	4051*	4067*	4078*	4087*	4110*	4163*	4215*	4257*	4303*	4348*	4387*	4420*						
		4448*	4656*	4755																
		3776	4576	4657	4742	4747#														
SCOPE	016320	2718#																		
SERFL	000772	2655#																		
SERNUM	000602	2727#	2856*	2994*	3003	3125														
SKAT	001014	2698#																		
SLAF	000722	2683#	2909	2911	2941*	2950	3123	3124	3942	4787	4810									
SLVN	000664	2632#	3157																	
SN	000540	5096	5099	5105	5107	5109#														
SNPG	020176	3158	5090#																	
SNPT	020120	2699#																		
SSCF	000724	2603	2844#																	
START	001330	2726#	2918	2920	4478															
STATC	001012	2723#																		
STATF	001004	4158	4214	4255	4301	4346	4476#													
STATIC	014574	2705#	2937*	2966	2971*	3005	3127	3394	3418	3441	3465	4476								
STFLG	000740	2940	2971#																	
STSCD	002466	2889#	2892																	
STO	001662	2895#	2897																	
ST1	001704	2607	2927#																	
ST2	002066	2650#	2849	2853*	2857*	2938	2962	2968	2991	3007	3028	3129	3159	3392						
SWR	000570	3416	3439	3463	4502	4510	4525	4561	4567	4572	4638	4648	4653	4680						
		4691	4696	4706	4725	4733	4738	4748	4759	4870	4872*	4879*	5118	5123						
		5127																		
SWREG	000176	2599#	2853	2857	4870	4879	5118													
TADX	001324	2837#																		
TC	000542	2633#	2941*	3146*	3222*	3223	3304*	3344*	3523*	3576*	3601*	3657*	3685*	3752*						

DTBOOT	1244#		
GETANS	768#		
LDPDR	516#		
LPDP11	1269#		
PSPTAG	747#		
REGBOX	133#		
RESLDR	874#		
SAVLDR	856#		
SVTK\$	1142#		
\$CATCH	1125#	2506#	2588
\$CHAIN	90#	2506#	2857
\$CHNMO	106#	2506#	2941
\$CNV16	607#		
\$CNV18	636#		
\$CNV48	705#		
\$CPCHK	898#		
\$CPREG	18#		
\$CPVEC	168#		
\$FPREG	47#		
\$GETAN	772#		
\$KMPRE	348#		
\$KWADR	999#		
\$KW11	930#		
\$LCTRL	3#		
\$LPREG	187#		
\$MAMFO	1177#		
\$MPBIT	208#		
\$MPREG	265#		
\$PDRBI	386#		
\$POWER	440#		
\$PSWBI	148#		
\$RECO	796#		
\$RESLD	877#		
\$RESTO	477#	2506#	5135
\$SAVE	465#	2506#	5134
\$SAVLD	859#		
\$SETTB	509#		
\$SHIFT	490#		
\$SMPRE	310#		
\$STINS	9#		
\$STKPT	203#		
\$ST200	1137#		
\$SVTK	1147#		
\$SWOPT	57#		
\$TCDRV	1030#		
\$TCREG	193#		
\$TRAPS	403#		
\$TYPE	519#		
\$TYPEF	593#		
\$UMPRE	273#		
\$VECTA	1164#		
.\$ACT1	68#	2506#	2589
.\$EOP	79#	2506#	2988

. ABS. 030644 000

TM03/T045 CONTROL LOGIC TEST PART I
CZTUOB.P11 13-MAY-80 14:47

C 11
MACY11 30A(1052) 13-MAY-80 14:50 PAGE 91-1
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0132

ERRORS DETECTED: 0

CZTUOB,CZTUOB/CRF=CZTUOB.P11
RUN-TIME: 18 31 3 SECONDS
RUN-TIME RATIO: 85/54=1.5
CORE USED: 15K (29 PAGES)