

TE16
TU77

TM03/TE16, TU77 CLT1
CZTEREO

COPYRIGHT (c) 1977-84
AH-A792E-MC
FICHE 01 OF 01

JUL 1984
digital
Made In USA



.REM •

IDENTIFICATION

PRODUCT CODE: AC-A791E-MC
PRODUCT NAME: CZTEAEO TMO3-TE16/TU77 CONTROL LOGIC TEST PART I
DATE CREATED: 15 MARCH 1984
MAINTAINER: TAPE DIAGNOSTIC ENGINEERING
AUTHOR: J. MITT

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

COPYRIGHT (©) 1977, 1984 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 TMO3. 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 TMO3 THE STRUCTURE AND ITS MAINTAINENCE MODES.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP-11 PROCESSOR
B. 8K OF CORE
C. CONSOLE TTY
D. TMO3 MAGTAPE CONTROLLER
E. MASSBUS CONTROLLER (RM)
F. TE16 MAGTAPE TRANSPORT

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE.

4. STARTING PROCEDURE

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

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

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

4.1 AUTOMATIC MODE OPERATION

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

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

**NOTE: THIS PROGRAM 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).

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

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

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

REGISTER START: <172440> (CR) [REGS:]
 VECTOR ADDRESS: <224> (CR) [VECT:]
 IS CONTROLLER JUMPERED IN NON-STANDARD MODE
 TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD <3> [JUMPER:]
 TM03 DRIVE: <0> (CR) [DRVN:]
 TE16 SLAVE: <0> (CR) [SLVN:]
 STATIC TESTS ONLY: <0> (CR) [STATC:]
 SLAVE TYPE (0=TE16,1=TU77): <0> (CR) [SLVTYP:]
 IF THE SOFTWARE SWR IS INVOKED:
 SWR = <000000> NEW = (CR) [SWREG:]

5. CONSOLE SWITCH SETTINGS

CONTROL:

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

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

SW15: 1=HALT ON ERROR
0=CONTINUE
SW14: 1=LOOP ON ERROR (SCOPE)
0=CONTINUE
SW13: 1=DO NOT PRINT ERRORS
0=PRINT ALL ERRORS
SW12: 1=HALT AT END OF PASS
0=DO CONTINUOUS CYCLE
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 (M8909 OR RH)
NON-EXIST DRIVE 3 EXPT-NOT RECVD
ITER: 3

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

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

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

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

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

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

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

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

LOGIC TEST 24: DPAR (M8906 RH)
DPAR EXPT EXPT-NOT RCVD

CS1	WC	BA	FC	CS2	DS	ER	AS	MR	TC
004260	000000	033726	000000	000100	010600	000000	000000	177712	140300

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

LOGIC TEST 26: FCE (M8909)
ERR NOT SET

CS1	WC	BA	FC	CS2	DS	ER	AS	MR	TC
004260	000000	001376	000000	000100	110600	001000	000001	000000	100300

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

LOGIC TEST 30: DTE (M8906 RH)
UNEXPECTED ERROR BITS

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

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

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

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

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

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 TMO3 AT THE ADDRESSES SPECIFIED BY THE OPERATOR. TEST OCCURS IMMEDIATELY AFTER DRIVE SELECTION.

PROGRAMMED SEQUENCE: FOR EACH TMO3 ADDRESS (0-7) THE C1 REGISTER IS READ, AND THE NON-EXISTANT DRIVE (NED) BIT IS CHECKED. NED IS SET WHEN THE TMO3 DOES NOT RESPOND TO DEM BY ISSUING TRA. IN THIS TEST, NED IS EXPECTED FOR EACH ADDRESS NOT TYPED BY THE OPERATOR.

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8909

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

LOGIC TEST #2: REGISTER ADDRESSING

PURPOSE: CHECK THE REGISTER SELECT LINES

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

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8909,M8905-YB,M8933

CIRCUITS	PRINT REFERENCE
C-LINES	(MB1,2,3),(MBI3),(MBI4),(MBI5)
RH REGISTER SELECT	(BCTA)
TMO3 REGISTER SELECT	(MBI2)
MASSBUS REGISTER SELECT LINES	(MB1,2)
PARITY TREE	(MBI4)
CPAR,ILR BITS	(MBI11)

LOGIC TEST #3: CONTROL BUS

PURPOSE: VERIFY THAT ALL CONTROL LINES PROPERLY TRANSMIT

ONES AND ZEROS.

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

LIKELY FAULT LOCATIONS: M5904,CABLES,M5903YA,M8909,M8905-YB,M8933

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

LOGIC TEST #4: SLAVE ADDRESSING

PURPOSE: VERIFY THE FUNCTIONING OF THE SLAVE ADDRESS BITS IN THE TAPE CONTROL REGISTER THE SLAVE ADDRESS BUS LINES, THE ADDRESS DECODE CIRCUIT IN THE TE16 AND THE SPR BIT.

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

PROGRAMMED SEQUENCE: THE SLAVE 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,M8910,M9001YA,M8933

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

LOGIC TEST #5: MAINTENANCE REGISTER BITS

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

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

LIKELY FAULT LOCATIONS: M8905-YB

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

LOGIC TEST #6: TAPE CONTROL REGISTER BITS

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

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

LIKELY FAULT LOCATIONS: M8909,M8905-YB

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

LOGIC TEST #7: FRAME COUNT BIT TEST

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

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

LIKELY FAULT LOCATIONS: M8909

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

LOGIC TEST #10: FUNCTION CODE BIT TEST

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

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

LIKELY FAULT LOCATION: M8909, M8905-YB

CIRCUITS -----	PRINT REFERENCE -----
TM03 REGISTER SELECTION	(MBI2)
FUNCTION CODE FLOPS	(MBI5)
FUNCTION CODE MULTIPLEXERS	(MR6)

LOGIC TEST #11: GO BIT SET, RESET

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

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

LIKELY FAULT LOCATION: MASSBUS CABLE B(INIT),M8909,M8905-YB

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

LOGIC TEST #12: DRIVE READY BIT

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

PRINT REF TCCM7

LOGIC TEST #13: INTERRUPT TEST

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

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

LIKELY FAULT LOCATION:

CIRCUITS PRINT REFERENCE

INTERRUPT CONTROL BCTF

MANUAL INTERVENTION TESTS 14,15,16,17

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

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

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

LIKELY FAULT LOCATION: M8910,SLAVE CABLE, M8933

CIRCUIT PRINT REFERENCE

MOL	LAW6, TCCM7, M8908, M9001YA, YC
WRL	LAW8, TCCM7, M8908, M9001YA, YC
DPR	TCCM7
DRY	TCCM7
BOT	LAW6, TCCM7, M8908YA, M8913, YA

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

PURPOSE: TO TEST ATA,DPR,DRY,SSC

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

LIKELY FAULT LOCATION: M8910,M8933,M8909,SLAVE CABLE

CIRCUIT

PRINT REFERENCE

SSC
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: M8910,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: M8909

CIRCUIT

PRINT REFERENCE

SET ILF DECODE
ILF FLOP
ILF MULTIPLEXER

MBI5,MBI7
MBI11
MBI10

LOGIC TEST #21: REGISTER MODIFICATION, REFUSED

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

LIKELY FAULT LOCATION: M8909

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

LOGIC TEST #22: CONTROL BUS PARITY (CPAR)

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

LIKELY FAULT LOCATIONS: M8909

CIRCUIT	PRINT REFERENCE
-----	-----
MASSBUS PARITY TREE	MBI4
CPAR FLOP	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,M8909

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

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

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

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

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

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

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

LIKELY FAULT LOCATION: M8909

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

LOGIC TEST #26: FRAME COUNT ERROR

PROGRAMMED SEQUENCE: SET WC TO -10. FC TO -20 WAM3 IN

MAINTENANCE REGISTER, LOAD WRITE AND GO, DELAY ISSUE MM OR
CLEAR. CHECK FCE AND CHECK FOR UNEXPECTED ERRORS. FRAME
COUNT ERROR SHOULD BE SET BECAUSE A WRITE OPERATION WAS
TERMINATED PRIOR TO A WORD COUNT OVERFLOW.

LIKELY FAULT LOCATIONS: M8909, MB CABLE, M8933, M8905-YB

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

LOGIC TEST #27: ILLEGAL REGISTER

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

LIKELY FAULT LOCATIONS: MASSBUSS, M8909

CIRCUITS -----	PRINT REFERENCE -----
REGISTER SELECT LINES	MB1, MB2
REGISTER SELECT DECODE	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: M8909, 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, M8909

CIRCUITS

PRINT REFERENCES

OPI TIMER
OPI FLOP
OPI TIMER CONTROL

TCCM5
MBI11
MBI7

LOGIC TEST 32: UNSAFE (UNS)

PROGRAMMED SEQUENCE:

A NON-EXISTANT SLAVE IS SELECTED AND A READ COMMAND IS ISSUED. UNSAFE ERROR IS CHECKED. IF THE DRIVE TYPE REG INDICATES A TU77 THEN NON-EXECUTABLE FUNCTION (NEF) IS ALSO CHECKED.

LIKELY FAULT LOCATIONS: M8909, M8910, SLAVE CABLE

CIRCUITS

PRINT REFERENCES

UNSAFE FLOP
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: M8909, M8933

CIRCUITS

PRINT REFERENCES

SPACE FUNCTION DECODE	MBI5
PIP GENERATION	TCCM7
STATUS REGISTER	TCCM7

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905-YB, SLAVE BUS, M8931, M8933

CIRCUITS

PRINT REFERENCES

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

LOGIC TEST 35: TAPE CONTROL WRITE (TCW)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATION: M8905-YB

CIRCUIT -----	PRINT REFERENCES -----
TCW	MR6

LOGIC TEST 36: FRAME COUNTER STATUS (FCS)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8909, M8933

CIRCUITS -----	PRINT REFERENCES -----
FCS BIT	MBI8
FCS MULTIPLEXER	TCCM7

M2

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

LOGIC TEST 42: CYCLIC REDUNDANCY ERROR

PROGRAMMED SEQUENCE:

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

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

CIRCUITS	PRINT REFERENCES
-----	-----
MM FUNCTION DECODE	MR5
CRC CHECK CIRCUIT	CNRZ3

LOGIC TEST 43: LRC

PROGRAMMED SEQUENCE:

A WRITE OPERATION IS PERFORMED WITH A MM FUNCTION (INC TMRL)
WHICH ASSERTS WD(SB) 5L THROUGHOUT THE RECORD. ALL ONES
DATA IS USED SO THAT THE FUNCTION ONLY INTERFERES WITH
THE WRITING OF THE LRC CHARACTER WHEN NONE OF THE TMO3
WRITE DATA LINES SHOULD BE ASSERTED.
** NOTE: THIS TEST IS NOT PERFORMED ON A TU77 SLAVE.

LIKELY FAULT LOCATIONS: M8505, M8933, M8910, M8934

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

LOGIC TEST 44: PE CORRECTABLE DATA
-----PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905-YB, M8901, M8932

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

LOGIC TEST 45: PE INCORRECTABLE DATA

REPEAT OF TEST 44, EXCEPT THAT THE MAINT. MODE FUNCTION GROUND 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, PCF.	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: M8909

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.

1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250

```

.LIST BIN,LOC,SEQ
.TITLE CZTEAEO TMO3-TE16/TU77 CTL I
:CONTROL LOGIC TEST PART I
:AC-A791E-MC
:FEB 77
:J.G. ADAMS
:REVISED MAY 1978 BY J. G. ADAMS ;..B CHANGED MODULE REFERENCES TO
;..B REFLECT TMO3 MODULES
;..B ADDED TU77 TEST CAPABILITY
:REVISED NOV 1978 BY M. PAGE ;. INDICATES ENHANCEMENTS TO
THE ORIGINAL REV (DZTEAA)
NECESSARY FOR TMO3
:REVISED MAY,1983 BY B. LEBLANC ;BL FIXED AIDS #CC0001220
:REVISED MARCH 15,1984 BY J. HITT ;ADD XON/XOFF FUNCTIONALITY
;MAKE TEXT CHANGES

NON-STANDARD JUMPER MODE (SEE 4.2 IN DOC.)
.MCALL ., $ACT11,.$EOP,$CATCH,$SAVE,$RESTORE,$CHAIN,$CHNMODE
.NLIST MC
.LIST ME
.ENABLE ABS,AMA

:CONSOLE SWITCHES*****
:
:SW15: 1=HALT ON ERROR
: 0=CONTINUE
:SW14: 1=LOOP ON ERROR
: 0=CONTINUE
:SW13: 1=DO NOT PRINT ERRORS
: 0=PRINT ERRORS
:SW12: 0=CONTINUOUS CYCLE
: 1=HALT AT END OF PASS
: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

```



```

1299                                     ;REGISTER EQUIVS*****
1300
1301          000000                      R0=#0
1302          000001                      R1=#1
1303          000002                      R2=#2
1304          000003                      R3=#3
1305          000004                      R4=#4
1306          000005                      R5=#5
1307          000006                      SP=#6
1308          000007                      PC=#7
1309
1311                                     ;ACT11 HOOK *****
(1)                                     ;SAVE CURRENT LOCATION CTR
(1)          000764                      $SVPC=.
(1)          000042                      . =42
(1) 000042 000000                      .WORD 0
(1)          000046                      . =46
(1) 000046 002666                      .WORD $ENDAD      ;SET LOCATION 46
(1)          000052                      . =52
(1) 000052 000000                      .WORD 0          ;SET LOCATION 52 = 0
(1)          000764                      .=$SVPC          ;RESTORE LOCATION CTR
(1)
1312                                     ;TTY INTERRUPT VECTOR*****
1313
1314          000060                      . =60
1315 000060 017256                      .WORD TTINT     ;TTY INTERRUPT HEADER ADDRESS
1316 000062 000340                      .WORD 340      ;PRIORITY LEVEL 7
1317
1318                                     ;SOFTWARE SWITCH REGISTER*****
1319                                     ;USED IF HARDWARE SWR = 177777 OR NOT AVAILABLE
1320          000176                      . =176
1321 000176 000000                      SWREG: .WORD 0   ;SOFTWARE SWITCH REGISTER
1322
1323                                     ;START ADDRESS*****
1324          000200                      . =200
1325 000200 000137 001334                      JMP START      ;PROGRAM START
1326
1327                                     ;RESTART ADDRESS*****
1328          000210                      . =210
1329 000210 000137 002226                      JMP ST2
1330
1331                                     ;TM03 INTERRUPT VECTOR*****
1332
1333          000224                      . =224
1334 000224 017246                      MTINT          ;TAPE INTERRUPT HANDLER ADDRESS
1335 000226 000340
1336

```

```

1338
1339          000510          .-510
1340          ;MASS BUS REGISTER EQUIVS*****
1341
1342 000510 172440      C1: 172440
1343 000512 172442      WC: 172442
1344 000514 172444      BA: 172444
1345 000516 172446      FC: 172446
1346 000520 172450      CS: 172450
1347 000522 172452      DS: 172452
1348 000524 172454      ER: 172454
1349 000526 172456      AS: 172456
1350 000530 172460      CC: 172460
1351 000532 172462      DB: 172462
1352 000534 172464      MR: 172464
1353 000536 172466      DT: 172466
1354 000540 172470      SN: 172470
1355 000542 172472      TC: 172472
1356
1357          ;ILLEGAL FUNCTION CODES
1358
1359 000544 005405      ILFT: 5405
1360 000546 007415      7415
1361 000550 016423      16423
1362 000552 020437      20437
1363 000554 022443      22443
1364 000556 025447      25447
1365 000560 031455      31455
1366 000562 033465      33465
1367 000564 036473      36473
1368
1369          ;CONSTANTS*****
1370
1371 000566 177776      PSW: 177776      ;PROCESSOR STATUS
1372 000570 177570      SWR: 177570      ;SWITCH REGISTER
1373 000572 177560      TKS: 177560      ;TTY READER STATUS
1374 000574 177562      TKB: 177562      ;TTY READ BUFFER
1375 000576 177564      TPS: 177564      ;TTY PUNCH STATUS
1376 000600 177566      TPB: 177566      ;TTY PUNCH BUFFER
1377 000602 000020      ITAMT: 20        ;ITERATION AMOUNT
1378 000604 000224      VECT: 224        ;INTERRUPT VECTOR(RH)
1379 000606 172440      REGS: 172440     ;STARTING REGISTER ADDRESS

```


;FLAGS AND COUNTERS*****

1381			
1382			
1383	000610	000000	TOB: 0
1384	000612	000000	TIB: 0
1385	000614	000000	HDRFL: 0
1386	000616	000000	EMADDR: 0
1387	000620	000000	DRVN: 0
1388	000622	000000	TR00: 0
1389	000624	000000	TR01: 0
1390	000626	000000	TR02: 0
1391	000630	000000	TR03: 0
1392	000632	000000	TR04: 0
1393	000634	000000	TR05: 0
1394	000636	000000	TR06: 0
1395	000640	000000	TR07: 0
1396	000642	000000	TR10: 0
1397	000644	000000	TR11: 0
1398	000646	000000	TR12: 0
1399	000650	000000	TR13: 0
1400	000652	000000	TR14: 0
1401	000654	000000	TR15: 0
1402	000656	000000	NRZOF: 0
1403	000660	000000	SLVN: 0
1404	000662	000000	PFLG: 0
1405	000664	000000	RTRN: 0
1406	000666	000000	ERADD: 0
1407	000670	000000	TEMP1: 0
1408	000672	000000	TEMP2: 0
1409	000674	000000	TEMP3: 0
1410	000676	000000	ITCNT: 0
1411	000700	000000	SAV1: 0
1412	000702	000000	SAV2: 0
1413	000704	000000	SAV3: 0
1414	000706	000000	SCOLP: 0
1415	000710	000000	ITRLP: 0
1416	000712	000000	EXFL: 0
1417	000714	000000	ATAF: 0
1418	000716	000000	SLAF: 0
1419	000720	000000	SSCF: 0
1420	000722	000000	ERRF: 0
1421	000724	000000	ASF: 0
1422	000726	000000	SCF: 0
1423	000730	000000	TREF: 0
1424	000732	000000	PEXFL: 0
1425	000734	000000	STFLG: 0
1426	000736	000000	LTADD: 0
1427	000740	000000	T24FL: 0
1428	000742	000000	ADDFL: 0
1429	000744	000000	WAM: 0
1430	000746	000000	FUN: 0
1431	000750	000000	DATC: 0
1432	000752	000000	WTAD: 0
1433	000754	000000	DATAD: 0
1434	000756	000000	RDAD: 0
1435	000760	000000	W2FLG: 0
1436	000762	000000	DERFL: 0

1437 000764 000000
 1438 000766 000000
 1439 000770 000000
 1440 000772 000000
 1441 000774 000000
 1442 000776 000000
 1443 001000 000000
 1444 001002 000000
 1445 001004 000000
 1446 001006 000000
 1447 001010 000000
 1448 001012 000000
 1449 001014 000000
 1450 001016 000003
 1451 001020 000000
 1452
 1453
 1454
 1455 001022 000000
 1456 001024 000000
 1457 001026 000000
 1458 001030 000000
 1459
 1460
 1461
 1462 001032 000005
 1463 001034 000005
 1464 001036 000012
 1465 001040 000012
 1466 001042 000000
 1467 001044 000017
 1468 001046 000017
 1469 001050 000017
 1470 001052 000017
 1471 001054 000000

PREFL: 0
 SERFL: 0
 CRCNT: 0
 UDES: 0
 WPGFL: 0
 PATRN: 0
 STATF: 0
 RDRVF: 0
 RCDP: 0
 STATC: 0
 SLVTYP: .WORD 0
 SKAT: 0
 PCNTR: 0
 JUMPER: 3
 NONSTD: 0

;B INDICATES SLAVE TYPE (0/1 = TE16/TU77)

;PASS COUNTER
 ;INDICATOR FOR NON-STANDARD CONFIG.
 ;FLAG FOR NON-STANDARD CONFIG.

;EXPT WRAP STATUS*****

WCS1: 0
 WCS2: 0
 WDS: 0
 WER: 0

;CORE DUMP PATTERNS*****

WCDP2: 5
 5
 12
 12
 0
 WCDPO: 17
 17
 17
 17
 0

Line No.	Code	Value	Label
1473			
1474			
1475			
1476	001056	000000	TSTTBL: 0
1477	001060	000000	0
1478	001062	002736	LT1
1479	001064	002736	LT1
1480	001066	003212	LT2
1481	001070	003212	LT2
1482	001072	003416	LT3
1483	001074	003420	LT3IT
1484	001076	003600	LT4
1485	001100	003600	LT4
1486	001102	004202	LT5
1487	001104	004210	LT5IT
1488	001106	004372	LT6
1489	001110	004374	LT6IT
1490	001112	004506	LT7
1491	001114	004510	LT7IT
1492	001116	004622	LT10
1493	001120	004624	LT10IT
1494	001122	004746	LT11
1495	001124	004750	LT11IT
1496	001126	005172	LT12
1497	001130	005174	LT12IT
1498	001132	005366	LT13
1499	001134	005376	LT13IT
1500	001136	005470	LT14
1501	001140	005530	LT14IT
1502	001142	005600	LT15
1503	001144	005640	LT15IT
1504	001146	005710	LT16
1505	001150	005750	LT16IT
1506	001152	006022	LT17
1507	001154	006062	LT17IT
1508	001156	006134	LT20
1509	001160	006150	LT20IT
1510	001162	006276	LT21
1511	001164	006312	LT21IT
1512	001166	006442	LT22
1513	001170	006456	LT22IT
1514	001172	006566	LT23
1515	001174	006602	LT23IT
1516	001176	006716	LT24
1517	001200	006732	LT24IT
1518	001202	007242	LT25
1519	001204	007250	LT25IT
1520	001206	007374	LT26
1521	001210	007402	LT26IT
1522	001212	007610	LT27
1523	001214	007634	LT27IT
1524	001216	007726	LT30
1525	001220	007750	LT30IT
1526	001222	010244	LT31
1527	001224	010252	LT31IT
1528	001226	011074	LT32

K3

1529	001230	011110	LT32IT
1530	001232	011252	LT33
1531	001234	011266	LT33IT
1532	001236	011354	LT34
1533	001240	011370	LT34IT
1534	001242	011510	LT35
1535	001244	011524	LT35IT
1536	001246	011662	LT36
1537	001250	011676	LT36IT
1538	001252	012002	LT37
1539	001254	012016	LT37IT
1540	001256	012152	LT40
1541	001260	012166	LT40IT
1542	001262	012272	LT41
1543	001264	012306	LT41IT
1544	001266	012534	LT42
1545	001270	012572	LT42IT
1546	001272	013064	LT43
1547	001274	013122	LT43IT
1548	001276	013330	LT44
1549	001300	013356	LT44IT
1550	001302	013576	LT45
1551	001304	013624	LT45IT
1552	001306	014042	LT46
1553	001310	014070	LT46IT
1554	001312	014304	LT47
1555	001314	014320	LT47IT
1556	001316	014474	LT50
1557	001320	014510	LT50IT
1558	001322	014650	LT51
1559	001324	014664	LT51IT
1560	001326	002622	
1561	001330	000051	
1562	001332	000000	

TADX: .WORD TEND
 TLAST: .WORD 51
 \$CNTRLS: .WORD 0

:CONTAINS # OF TESTS
 :XON/XOFF FLAG


```

1564                                     .EVEN
1565                                     ;PROGRAM START AND HOUSEKEEPING*****
1566
1567                                     ;NOTE: PROGRAM STARTS HERE ON START AT 200
1568 001334 012706 000500 START: MOV #500,SP ;SET STACK POINTER
1569 001340 013746 000004 MOV @#4,-(SP) ;SAVE ERROR TRAP VECTOR
1570 001344 013746 000006 MOV @#6,-(SP) ;AND VECTOR +2
1571 001350 012737 001374 000004 MOV #1,@#4 ;SET NEW VECTOR
1572 001356 005037 000006 CLR @#6 ;AND PSW
1573 001362 022777 177777 177200 CMP #-1,@SWR ;USE SOFTWARE SWITCH IF HARDWARE
1574 001370 001402 BEQ 2$ ;IS = 177777
1575 001372 000404 BR 3$ ;OTHERWISE USE HARDWARE SWR
1576 001374 022626 1$: CMP (SP)+,(SP)+ ;RESET STACK PTR
1577 001376 012737 000176 000570 2$: MOV #SWREG,SWR ;SET SOFTWARE SWITCH REGISTER
1578 001404 012637 000006 3$: MOV (SP)+,@#6 ;RESTORE ERROR TRAP VECTORS
1579 001410 012637 000004 MOV (SP)+,@#4
1580 001414 005037 001012 CLR SKAT ;CLEAR SKIP ADDRESS TEST FLAG
1581 001420 005027 CLR (PC)+ ;:CLEAR CHAIN INDICATOR
(1) 001422 000000 CHNFLG: .WORD 0 ;:CHAIN MODE INDICATOR
(1) ;:1/0 = CHAIN/NOT CHAIN MODE
(1) 001424 005737 000042 TST @#42 ;:BRANCH IF IN DUMP MODE
(1) 001430 001407 BEQ 50$
(1) 001432 012737 000176 000570 MOV #SWREG,SWR ;:INVOKE SOFTWARE SWR
(1) 001440 005237 001422 INC CHNFLG ;:SET CHNFLG = CHAIN MODE
(1) 001444 000137 002246 JMP TSCD ;:GO TO CHAIN ADDRESS
(1) 50$:
1582 001450 000240 SCHN: NOP
1583 001452 122737 000006 000041 4$: CMPB #6,@#41 ;:BRANCH IF NOT LOADED VIA TMDP
1584 001460 001005 BNE 5$
1585 001462 012704 023546 MOV #MSG62,R4 ;ADVISE USER TO REMOVE TMDP FROM
1586 001466 004737 017760 JSR PC,TTOUT ;UNIT UNDER TEST
1587 001472 000000 HALT
1588 001474 012704 020762 5$: MOV #MSG1,R4
1589 001500 004737 017760 JSR PC,TTOUT ;PRINT TITLE
1590 001504 005737 001422 TST CHNFLG ;SEE IF IN CHAIN MODE
1591 001510 001402 BEQ 6$ ;IF NOT: BR
1592 001512 000137 002246 JMP TSCD ;ELSE GO TO START OF TESTS
1593 001516 112737 000043 020762 6$: MOVB #'#,MSG1 ;DO NOT PRINT TITLE ON RESTART
1594 001524 012704 022723 MOV #MSG44,R4
1595 001530 004737 017760 JSR PC,TTOUT ;REQUEST REGISTER ADDRESS
1596 001534 013703 000606 MOV REGS,R3
1597 001540 004737 020172 JSR PC,OCPT ;PRINT CURRENT ADDRESS
1598 001544 012705 000606 MOV #REGS,R5 ;SET ADDRESS SAVE LOC
1599 001550 012701 000007 MOV #7,R1 ;SET SIZE OF RESPONSE
1600 001554 012702 176400 MOV #176400,R2 ;SET UPPER LIMIT
1601 001560 012703 172300 MOV #172300,R3 ;SET LOWER LIMIT
1602 001564 004737 017436 JSR PC,ITR ;GO GET RESPONSE
1603 001570 012704 022745 MOV #MSG45,R4
1604 001574 004737 017760 JSR PC,TTOUT ;REQUEST VECTOR
1605 001600 013703 000604 MOV VECT,R3
1606 001604 004737 020172 JSR PC,OCPT ;PRINT CURRENT VECTOR
1607 001610 012705 000604 MOV #VECT,R5 ;SET ADDRESS SAVE LOC
1608 001614 012701 000004 MOV #4,R1 ;SET SIZE OF RESPONSE
1609 001620 012702 000224 MOV #224,R2 ;SET UPPER LIMIT
1610 001624 012703 000150 MOV #150,R3 ;SET LOWER LIMIT
1611 001630 004737 017436 JSR PC,ITR ;GO GET RESPONSE

```


1612	001634	013700	000604		MOV	VECT,R0	;GET VECTOR
1613	001640	012720	017246		MOV	#MTINT,(R0)+	;LOAD INTERRUPT ADDRESS IN VECTOR
1614	001644	012710	000340		MOV	#340,(R0)	;LOAD PRIORITY
1615	001650	013700	000606		MOV	REGS,R0	;GET START OF REGS
1616	001654	012701	000016		MOV	#16,R1	;SET NUMBER OF REGS
1617	001660	012702	000510		MOV	#C1,R2	;GET START OF TABLE
1618	001664	010022		STO:	MOV	R0,(R2)+	;BUILD TABLE
1619	001666	062700	000002		ADD	#2,R0	;BUMP ADDRESS
1620	001672	005301			DEC	R1	;SEE IF DONE
1621	001674	001373			BNE	STO	;IF NOT: BR
1622	001676	012702	000610		MOV	#TOB,R2	
1623	001702	012700	000077		MOV	#77,R0	
1624	001706	005022		ST1:	CLR	(R2)+	;CLEAR FLAGS + COUNTERS
1625	001710	005300			DEC	R0	
1626	001712	001375			BNE	ST1	
1627	001714	012704	023356		MOV	#MS57A,R4	;+REQUEST IF JUMPER IS IN NON-STANDARD MODE
1628	001720	004737	017760		JSR	PC,TTOUT	;+
1629	001724	012705	001016		MOV	#JUMPER,R5	;+
1630	001730	012703	000000		MOV	#0,R3	;+LIMIT RESPONSE
1631	001734	012701	000002		MOV	#2,R1	;+SET CHAR. NUMBER TO 1
1632	001740	012702	000004		MOV	#4,R2	;+SET RANGE 0 - 4
1633	001744	004737	017436		JSR	PC,TTR	;+GET RESPONSE
1634	001750	022737	000002	001016	CMP	#2,JUMPER	;TEST FOR NON-STANDARD JUMPER.
1635	001756	001002			BNE	1\$	
1636	001760	004737	016670		JSR	PC,NOST	;GO TO MODIFY SCHEDLAR
1637	001764	012704	023340	1\$:	MOV	#MSG57,R4	;REQUEST TM03 DRIVE #
1638	001770	004737	017760		JSR	PC,TTOUT	
1639	001774	013703	000620		MOV	DRVN,R3	;GET CURRENT DRIVE #
1640	002000	004737	020172		JSR	PC,OCTP	;PRINT IT
1641	002004	012705	000620		MOV	#DRVN,R5	;TTR ROUTINE RETURNS USER VALUE TO (R5)
1642	002010	012701	000002		MOV	#2,R1	;LIMIT RESPONSE
1643	002014	012702	000007		MOV	#7,R2	;LIMIT RANGE TO 0-7
1644	002020	012703	000000		MOV	#0,R3	
1645	002024	004737	017436		JSR	PC,TTR	;GET USER RESPONSE
1646	002030	012704	023517		MOV	#MSG58,R4	;REQUEST TE16 SLAVE #
1647	002034	004737	017760		JSR	PC,TTOUT	
1648	002040	013703	000660		MOV	SLVN,R3	;GET CURRENT SLAVE #
1649	002044	004737	020172		JSR	PC,OCTP	;AND PRINT IT
1650	002050	012705	000660		MOV	#SLVN,R5	;TTR ROUTINE RETURNS RESPONSE TO (R5)
1651	002054	012701	000002		MOV	#2,R1	;LIMIT RESONSE TO 1 CHARACTER
1652	002060	012702	000007		MOV	#7,R2	;BETWEEN 0 AND 7
1653	002064	012703	000000		MOV	#0,R3	
1654	002070	004737	017436		JSR	PC,TTR	;GET USER RESPONSE
1655	002074	012704	023313		MOV	#MSG56,R4	
1656	002100	004737	017760		JSR	PC,TTOUT	;REQUEST STATIC ONLY
1657	002104	013703	001006		MOV	STATC,R3	;GET CURRENT VALUE
1658	002110	004737	020172		JSR	PC,OCTP	;AND TYPE IT
1659	002114	012705	001006		MOV	#STATC,R5	;SET ADDRESS OF STATIC FLAG
1660	002120	012701	000002		MOV	#2,R1	;SET SIZE OF RESPONSE
1661	002124	012702	000001		MOV	#1,R2	;SET UPPER LIMIT
1662	002130	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
1663	002134	004737	017436		JSR	PC,TTR	;GET RESPONSE
1664							
1665	002140	012704	023672		MOV	#MSG67,R4	;+B REQUEST SLAVE TYPE 'TE16 OR TU77'
1666	002144	004737	017760		JSR	PC,TTOUT	;+B
1667	002150	013703	001010		MOV	SLVTYP,R3	;+B GET CURRENT SLAVE TYPE

N3

CZTEAEO TM03-TE16/TU77 CTL I
CZTEAE.P11 06-APR-84 09:45

MACY11 30(1046) 06-APR-84 09:48 PAGE 34-2

SEQ 0039

1668	002154	004737	020172		JSR	PC,OCTP	;++B TYPE CURRENT VALUE
1669	002160	012705	001010		MOV	#SLVTYP,R5	;++B GET ADDRESS OF SLVTYP FLAG
1670	002164	012701	000002		MOV	#2,R1	;++B SET SIZE OF RESPONSE
1671	002170	012702	000001		MOV	#1,R2	;++B SET UPPER LIMIT
1672	002174	012703	000000		MOV	#0,R3	;++B SET LOWER LIMIT
1673	002200	004737	017436		JSR	PC,ITR	;++B GET RESPONSE
1674	002204	005737	001010		TST	SLVTYP	:IS IT A TU77
1675	002210	001406			BEQ	ST2	:BRANCH IF NOT
1676	002212	012737	116741	006020	MOV	#116741,STWD16	:SET UP TEST WORD FOR TEST 16
1677	002220	012737	110741	006132	MOV	#110741,STWD17	:SET UP TEST WORD FOR TEST 17
1678							
1679							
1680							
1681	002226	012706	000500		:START 210		
1682	002232	005037	001002		ST2: MOV	#500,SP	:SET STACK PTR
1683	002236	005037	001014		CLR	RDRVF	:CLEAR REVERSE FLAG
1684	002242	004737	020620		CLR	PCNTR	:CLEAR PASS COUNTER
					JSR	PC,GTSWR	:GET SWITCHES

```

1686
1687
1688
1689 002246 052777 000100 176316 TSCD: BIS @100,@TKS ;SET KEYBOARD INTERRUPT ENABLE
1690 002254 005737 000042 TST @042 ;ACT MODE ?
1691 002260 001407 BEQ 1$ ;BRANCH IF NOT
1692 002262 032737 000004 172466 BIT @4,@0172466
1693 002270 001403 BEQ 1$
1694 002272 012737 000001 001010 MOV @1,SLVTYP
1695 002300 005037 000774 1$: CLR WPGFL ;CLEAR WRAP PATRN FLAG
1696 002304 005037 000734 CLR STFLG ;CLEAR SINGLE TEST FLAG
1697 002310 017700 176254 MOV @SWR,RO
1698 002314 042700 177700 BIC @177700,RO ;BRANCH IF SINGLE
1699 002320 001122 BNE STSCD ;TEST SELECTED
1700 002322 005737 001422 TST CHNFLG ;:BRANCH IF NOT IN CHAIN MODE
(1) 002326 001457 BEQ TSCDA
(1) 002330 012737 177777 000620 MOV @-1,DRVN ;; INITIALIZE DRIVE #
(1) 002336 012737 177777 000660 NxtDRV: MOV @-1,SLVN ;; INITIALIZE SLAVE #
(1) 002344 012777 000040 176146 1$: MOV @40,@CS ;; INIT CONTROLLER
(1) 002352 005237 000620 INC DRVN ;; STEP DRIVE #
(1) 002356 022737 000010 000620 CMP @10,DRVN ;; EXIT IF ALL DRIVES TESTED
(1) 002364 001521 BEQ $DONE ;; FOR AVAILABILITY
(1) 002366 013777 000620 176124 MOV DRVN,@CS ;; LOAD DRIVE #
(1) 002374 005777 176110 TST @C1 ;; ACCESS DRIVE
(1) 002400 032777 010000 176112 BIT @10000,@CS ;; BRANCH IF DRIVE NON EXISTANT
(1) 002406 001356 BNE 1$ ;; (NED = 1)
(1) 002410 005237 000660 NxtSLV: INC SLVN ;; STEP SLAVE # AND BRANCH
(1) 002414 001011 BNE 1$ ;; IF NOT SLAVE 0
(1) 002416 005737 000620 TST DRVN ;; BRANCH IF NOT DRIVE # 0
(1) 002422 001006 BNE 1$
(1) 002424 122737 000006 000041 CMPB @6,@041 ;; BRANCH IF NOT TMDP
(1) 002432 001002 BNE 1$
(1) 002434 005237 000660 INC SLVN ;; STEP TO SLAVE # 1
(1) 002440 022737 000010 000660 1$: CMP @10,SLVN ;; BRANCH IF ALL SLAVES TESTED
(1) 002446 001733 BEQ NxtDRV ;; FOR AVAILABILITY
(1) 002450 013777 000660 176064 MOV SLVN,@C ;; LOAD SLAVE UNIT #
(1) 002456 032777 002000 176052 BIT @2000,@DT ;; BRANCH IF SLAVE NOT
(1) 002464 001751 BEQ NxtSLV ;; PRESENT (SPR = 0)
1701 002466 012737 001056 000736 TSCDA: MOV @TSTTBL,LTADD
1702 002474 062737 000004 000736 TSCD0: ADD @4,LTADD
1703 002502 013737 000736 000710 TSCD1: MOV LTADD,IIRLP
1704 002510 062737 000002 000710 ADD @2,IIRLP ;SET ITERATION ADDRESS
1705 002516 005037 000614 CLR HDRFL ;CLEAR PRINT HEADER FLAG
1706 002522 017700 176210 MOV @LTADD,RO ;SET POINTER TO TEST
1707 002526 000110 JMP (RO) ;GO TO TEST
1708 002530 032777 002000 176032 TSCD2: BIT @2000,@SWR ;SEE IF HALT ON TEST
1709 002536 001403 BEQ TSCD3 ;IF NOT: BR
1710 002540 000000 HALT
1711 002542 005037 000774 CLR WPGFL ;CLEAR WRAP DATA GENERATOR FLAG
1712 002546 005737 000734 TSCD3: TST STFLG ;SE IF SINGLE TEST
1713 002552 001750 BEQ TSCD0 ;IF NOT: BR
1714 002554 017700 176010 MOV @SWR,RO ;BRANCH IF ALL TESTS DESIRED
1715 002560 042700 177700 BIC @177700,RO ;IF SO: BR
1716 002564 001630 BEQ TSCD ;SET SINGLE TEST FLAG
1717 002566 012737 000001 000734 STSCD: MOV @1,STFLG ;SEE IF EXCEEDED TESTS
1718 002574 023700 001330 CMP TLAST,RO

```



```

1719 002600 002410          BLT      TEND          ;IF 50: BR
1720 002602 006300          ASL      RO
1721 002604 006100          ROL      RO          ;SET TABLE MODIFIER
1722 002606 012737 001056 000736  MOV     @TSTTB,LTADD
1723 002614 060037 000736  ADD     RO,LTADD     ;SET TEST POINTER
1724 002620 000730          BR       TSCD1
1725 002622 005737 001422  TEND:  TST     CHNFLG     ;BRANCH IF IN CHAIN MODE
1726 002626 001270          BNE     NXTSLV     ;STEP TO NEXT SLAVE
1727 002630 012704 022563  $DONE: MOV     @MSG41,R4
1728 002634 004737 017760  JSR     PC,TTOUT    ;PRINT END OF PASS
1729 002640 013703 001014  MOV     PCNTR,R3
1730 002644 004737 020172  JSR     PC,OCTP
1731 002650 005000          CLR     RO          ;PRINT PASS NUMBER
1732 002652 005300          1$:    DEC     RO          ;DELAY WAITING FOR
1733 002654 001376          BNE     1$
1734 002656 013700 000042  MOV     @#42,RO     ;GET ACT11 RETURN ADDRESS
(1) 002662 001405          BEQ     HERE
(1) 002664 000005          RESET
(1) 002666 004710          $ENDAD: JSR     PC,(RO)
(1) 002670 000240          NOP
(1) 002672 000240          NOP
(1) 002674 000240          NOP
(1) 002676 000240          HERE:  NOP
1735 002700 005737 001422  TST     CHNFLG     ;BRANCH IF IN CHAIN MODE
1736 002704 001005          BNE     TENDX
1737 002706 032777 010000 175654  BIT     @10000,@SWR ;SEE IF HALT ON PASS
1738 002714 001401          BEQ     TENDX     ;IF NOT: BR
1739 002716 000000          HALT
1740 002720 012737 000001 001012  TENDX: MOV     @1,SKAT  ;SET SKIP ADDRESS TEST FLAG
1741 002726 005237 001014  INC     PCNTR
1742 002732 000137 002246  JMP     TSCD
1743          ;LOGIC TEST 1: DRIVE ADDRESSING*****
1744
1745 002736 012737 024012 000616  LT1:   MOV     @MSLT1,EMADDR ;.B SET ERROR MSG HDR ADDRESS
1746 002744 013737 000620 000674  MOV     DRVN,TEMP3  ;GET DRIVE # TO BE TESTED
1747 002752 013701 000620          MOV     DRVN,R1
1748 002756 005737 001012          TST     SKAT
1749 002762 001403          BEQ     1$        ;SEE IF SKIP ADDRESS TESTS
1750 002764 005737 000734          TST     STFLG
1751 002770 001506          BEQ     LT1X     ;IF NOT: BR
1752 002772 032777 001000 175570  1$:    BIT     @1000,@SWR ;SEE IF SINGLE TEST
1753 003000 001430          BEQ     LT1A     ;BRANCH IF MAN INTERVENTION
1754 003002 012704 021226          LT1G0: MOV     @MSG2A,R4 ;NOT SELECTED
1755 003006 004737 017200          JSR     PC,INST
1756 003012 012704 021165          LT1G:  MOV     @MSG2,R4 ;PRINT TEST INSTRUCTIONS
1757 003016 004737 017760          JSR     PC,TTOUT
1758 003022 012705 000674          MOV     @TEMP3,R5 ;REQUEST DRIVE NUMBER
1759 003026 012701 000002          MOV     @2,R1    ;TTR ROUTINE RETURNS RESPONSE TO (R5)
1760 003032 012702 000007          MOV     @7,R2
1761 003036 012703 000000          MOV     @0,R3
1762 003042 004737 017436          JSR     PC,TTR
1763 003046 005737 000670          TST     TEMP1
1764 003052 001455          BEQ     LT1X     ;GET DRIVE NUMBER
1765 003054 005001          CLR     R1        ;SEE IF ANOTHER DRIVE
1766 003056 012700 000010          MOV     @10,RO   ;IF NOT: BR
1767 003062 012777 000040 175430  LT1A:  MOV     @40,@CS   ;SELECT DRIVE 0
          ;SET NUMBER OF DRIVES
          ;INIT

```

1768	003070	010177	175424			MOV	R1,@CS	;SELECT DRIVE
1769	003074	005777	175410			TST	@C1	;ACCESS DRIVE
1770	003100	032777	010000	175412		BIT	@10000,@CS	;SEE IF NED
1771	003106	001010				BNE	LT1B	;IF SO: BR
1772	003110	032777	001000	175452		BIT	@1000,@SWR	;BRANCH IF NOT MANUAL INTERVENTION
1773	003116	001433				BEQ	LT1X	
1774	003120	023701	000674			CMP	TEMP3,R1	;SEE IF SHOULD BE NED
1775	003124	001404				BEQ	LT1C	;IF NOT: BR
1776	003126	000407				BR	LT1ER	;ELSE GO TO ERROR
1777	003130	023701	000674		LT1B:	CMP	TEMP3,R1	;SEE IF SHOULD BE NED
1778	003134	001410				BEQ	LT1ER1	
1779	003136	005300			LT1C:	DEC	RO	
1780	003140	001724				BEQ	LT1G	;IF DONE ALL: BR
1781	003142	005201				INC	R1	;SELECT NEXT DRIVE
1782	003144	000746				BR	LT1A	;CONTINUE
1783	003146	012737	000001	000712	LT1ER:	MOV	@1,EXFL	;FLAG EXPT
1784	003154	000403				BR	LT1ER2	
1785	003156	012737	000002	000712	LT1ER1:	MOV	@2,EXFL	;FLAG NOT EXPT
1786	003164	012737	021355	000666	LT1ER2:	MOV	@MSG3,ERADD	;FLAG CONDITION
1787	003172	012737	003062	000706		MOV	@LT1A,SCOLP	;SET SCOPE ADDRESS
1788	003200	004737	015230			JSR	PC,LTGER	;GO PRINT LOGIC TEST ERROR
1789	003204	000754				BR	LT1C	;CONTINUE TEST
1790	003206	000137	002530		LT1X:	JMP	TSCD2	;RETURN TO SCHED
1791								


```

1793                                     ;LOGIC TEST 2: REGISTER ADDRESSING*****
1794
1795 003212 000240                       LT2:  NOP
1796 003214 012777 000040 175276      LT2IT: MOV  #40,@CS          ;INIT
1797 003222 013777 000620 175270      MOV  DRVN,@CS        ;SELECT DRIVE
1798 003230 012737 024066 000616      MOV  #MSLT2,EMADDR  ;SAVE LT2 HEADER ADDRESS
1799 003236 012705 000510              MOV  #C1,R5         ;SET ADDRESS OF FIRST REGISTER
1800 003242 012700 000016              MOV  #16,R0        ;SET NUMBER OF REGISTERS
1801 003246 012702 000622              MOV  #TR00,R2      ;SET START OF REGISTER BUFFER
1802 003252 011501                       LT2A: MOV  (R5),R1
1803 003254 011112                       MOV  (R1),(R2)     ;READ REGISTER
1804 003256 032777 020000 175224      BIT  #20000,@C1    ;SEE IF ERROR
1805 003264 001402                       BEQ  LT2B          ;IF NOT: BR
1806 003266 004737 003316              JSR  PC,LT2ER1     ;ELSE GO TO ERROR 1
1807 003272 032777 000002 175224      LT2B: BIT  #2,@ER   ;SEE IF ILR
1808 003300 001402                       BEQ  LT2C          ;IF NOT: BR
1809 003302 004737 003334              JSR  PC,LT2ER2     ;ELSE GO TO ERROR 2
1810 003306 022225                       LT2C: CMP  (R2)*,(R5)* ;BUMP ADDRESS
1811 003310 005300                       DEC  R0
1812 003312 001357                       BNE  LT2A          ;CONTINUE FOR ALL REGISTERS
1813 003314 000434                       BR   LT2X
1814
1815 003316 012737 000002 000712      LT2ER1: MOV #2,EXFL  ;FLAG NOT EXPECTED
1816 003324 012737 021377 000666      MOV  #MSG4,ERADD   ;POINT TO CONTROLLER ERROR
1817 003332 000415                       BR   LT2ERG        ;GO TO ERROR
1818 003334 012737 000002 000712      LT2ER2: MOV #2,EXFL  ;FLAG NOT EXPECTED
1819 003342 012737 021415 000666      MOV  #MSG5,ERADD   ;POINT TO DRIVE ERROR
1820 003350 000406                       BR   LT2ERG        ;GO TO ERROR
1821 003352 012737 000001 000712      LT2ER3: MOV #1,EXFL  ;FLAG EXPECTED
1822 003360 012737 021377 000666      MOV  #MSG4,ERADD   ;POINT TO DRIVE
1823 003366 012737 003402 000706      LT2ERG: MOV #LT2LP,SCOLP ;SET SCOPE ADDRESS
1824 003374 004737 015230              JSR  PC,LTGER      ;GO PRINT
1825 003400 000207                       RTS  PC            ;ELSE CONTINUE
1826 003402 005726                       LT2LP: TST (SP)*   ;RESET STACK
1827 003404 000722                       BR   LT2A          ;LOOP
1828 003406 004737 016600              LT2X: JSR PC,ITER   ;GO SEE IF ITERATIONS
1829 003412 000137 002530              JMP  TSCD2         ;RETURN TO SCHED

```

```

1831                                     ;LOGIC TEST 3: CONTROL BUS*****
1832
1833 003416 000240                       LT3:  NOP
1834 003420 012737 024145 000616  LT3IT: MOV    #MSLT3,EMADDR ;SET TEST HEADER
1835 003426 012701 000001             MOV    #1,R1          ;PRESET PATTERN 1
1836 003432 012700 000020             MOV    #20,R0         ;SET PATTERN CHANGE NUMBER
1837 003436 004737 016726             LT3A: JSR    PC,INIT1 ;GO INIT
1838 003442 010177 175050             MOV    R1,@FC        ;WRITE TO FC
1839 003446 032777 000010 175050     BIT    #10,@ER       ;SEE IF CPAR (TMO3)
1840 003454 001012                     BNE    LT3ER1        ;IF SO: BR
1841 003456 017702 175034             LT3B: MOV    @FC,R2   ;READ FC
1842 003462 032777 020000 175020     BIT    #20000,@C1    ;SEE IF MCPE (RH)
1843 003470 001017                     BNE    LT3ER2        ;IF SO: BR
1844 003472 005300                       LT3C: DEC    R0       ;SEE IF DONE PATTERN CHANGES
1845 003474 001426                       BEQ    LT3X          ;IF SO: BR
1846 003476 006301                       ASL    R1            ;CHANGE PATTERN
1847 003500 000756                       BR     LT3A          ;CONTINUE
1848 003502 012737 021730 000666  LT3ER1: MOV   #MSG11,ERADD ;SET ERROR CODE
1849 003510 012737 003436 000706     MOV   #LT3A,SCOLP    ;SET SCOPE ADDRESS
1850 003516 017702 174774             MOV   @FC,R2        ;GET DATA
1851 003522 004737 016336             JSR   PC,LTGER1     ;GO DO ERROR
1852 003526 000753                       BR     LT3B
1853 003530 012737 021704 000666  LT3ER2: MOV   #MSG10,ERADD ;SET ERROR CODE
1854 003536 012737 003456 000706     MOV   #LT3B,SCOLP    ;SET SCOPE ADDRESS
1855 003544 004737 016336             JSR   PC,LTGER1     ;GO DO ERROR
1856 003550 000750                       BR     LT3C
1857 003552 105701                       LT3X: TSTB   R1      ;SEE IF DONE PATTERN 2
1858 003554 100405                       BMI    LT3XX        ;IF SO: BR
1859 003556 012701 000401             MOV   #401,R1       ;SET PATTERN 2
1860 003562 012700 000010             MOV   #10,R0        ;SET PATTERN CHANGE NUMBER
1861 003566 000723                       BR     LT3A         ;DO PATTERN 2
1862 003570 004737 016600             LT3XX: JSR   PC,ITER ;GO SEE IF ITERATIONS
1863 003574 000137 002530             JMP    TSCD2        ;RETURN TO SCHEDULAR

```



```

1865
1866
1867
1868 003600 013737 000660 000674 LT4:  MOV    SLVN,TEMP3
1869 003606 013701 000660          MOV    SLVN,R1
1870 003612 005737 001012          TST    SKAT          ;SEE IF SKIP ADDRESS TESTS
1871 003616 001403          BEQ    1$           ;IF NOT: BR
1872 003620 005737 000734          TST    STFLG        ;SEE IF SINGLE TEST
1873 003624 001564          BEQ    LT4X         ;IF NOT: BR
1874 003626 032777 001000 174734 1$:  BIT    #1000,@SWR   ;BRANCH IF MAN INTERVENTION
1875 003634 001430          BEQ    LT4A         ;NOT SELECTED
1876 003636 012704 021533          LT4G0: MOV    #MSG8A,R4
1877 003642 004737 017200          JSR    PC,INST      ;PRINT TEST INSTRUCTIONS
1878 003646 012704 021472          LT4G:  MOV    #MSG8,R4
1879 003652 004737 017760          JSR    PC,TTOUT     ;REQUEST SLAVE
1880 003656 012705 000674          MOV    #TEMP3,R5
1881 003662 012701 000002          MOV    #2,R1
1882 003666 012702 000007          MOV    #7,R2
1883 003672 012703 000000          MOV    #0,R3
1884 003676 004737 017436          JSR    PC,TTR
1885 003702 005737 000670          TST    TEMP1        ;GET SLAVE NUMBER
1886 003706 001533          BEQ    LT4X         ;SEE IF SLAVE
1887 003710 005001          CLR    R1           ;IF NOT: BR
1888 003712 012700 000010          MOV    #10,R0       ;SELECT SLAVE 0
1889 003716 012777 000040 174574 LT4A:  MOV    #40,@CS      ;SET NUMBER OF SLAVES
1890 003724 013777 000620 174566          MOV    DRVN,@CS     ;INIT
1891 003732 010177 174604          MOV    R1,@TC       ;SELECT DRIVE
1892 003736 017703 174574          MOV    @DT,R3       ;SELECT SLAVE
1893 003742 003742 020137 000674          CMP    R1,TEMP3    ;GET DT
1894 003746 001404          BEQ    LT4B         ;SEE IF SHOULD HAVE SPR
1895 003750 032703 002000          BIT    #2000,R3    ;IF SO: BR
1896 003754 001461          BEQ    LT4D         ;SEE IF SPR
1897 003756 000464          BR     LT4ER1       ;IF NOT: BR
1898 003760 032703 002000          LT4B:  BIT    #2000,R3 ;GO TO ERROR 1
1899 003764 001465          BEQ    LT4ER2       ;SEE IF NO SLAVE PRESENT
1900 003766 012704 023642          LT4C:  MOV    #MSG64,R4  ;(SPR=0)
1901 003772 004737 017760          JSR    PC,TTOUT     ;TYPE 'SLAVE TYPE = '
1902 003776 012702 000001          MOV    #1,R2
1903 004002 012704 023660          MOV    #MSG65,R4   ;PRESET SLAVE TYPE = TU77
1904 004006 022777 142054 174522          CMP    #142054,@DT ;SET UP TO TYPE 'TU77'
1905 004014 001412          BEQ    1$           ;BRANCH IF TU77
1906 004016 012704 023665          MOV    #MSG66,R4   ;CHANGE SLAVE TYPE TO 'TE16'
1907 004022 005302          DEC    R2           ;CHANGE SLAVE TYPE TO TE16
1908 004024 022777 142051 174504          CMP    #142051,@DT ;BRANCH IF TE16
1909 004032 001403          BEQ    1$
1910 004034 005302          DEC    R2           ;CHENGE SLAVE TYPE TO ILLEGAL
1911 004036 012704 024002          MOV    #MSG69,R4
1912 004042 004737 017760          1$:  JSR    PC,TTOUT   ;TYPE SLAVE TYPE (TU77,TE16, OR ILLEGAL)
1913 004046 020237 001010          CMP    R2,SLVTYP  ;BRANCH IF HARDWARE SLAVE TYPE IS THE
1914 004052 001406          BEQ    4$           ;SAME AS USER SPACIFIED SLAVE TYPE
1915 004054 012704 023730          MOV    #MSG68,R4   ;..B TYPE 'INCORRECT SLAVE TYPE'
1916 004060 004737 017760          JSR    PC,TTOUT   ;..B
1917 004064 000137 002622          JMP    TEND        ;..B EXIT TEST
1918 004070 012704 022405          4$:  MOV    #MSG30,R4
1919 004074 004737 017760          JSR    PC,TTOUT   ;PRINT SERIAL NUMBER TAG
1920 004100 017703 174434          MOV    @SN,R3

```

1921	004104	004737	020516			JSR	PC,SNPT	;PRINT SERIAL NUMBER
1922	004110	032777	001000	174452		BIT	#1000,@SWR	;BRANCH IF NOT MANUAL INTERVENTION
1923	004116	001427				BEQ	LT4X	
1924	004120	005300			LT4D:	DEC	R0	
1925	004122	001651				BEQ	LT4G	;IF DONE ALL: BR
1926	004124	005201				INC	R1	;BUMP SLAVE
1927	004126	000673				BR	LT4A	;CONTINUE
1928	004130	012737	000001	000712	LT4ER1:	MOV	#1,EXFL	;FLAG EXPT: NOT RECEIVED
1929	004136	000403				BR	LT4ERG	
1930	004140	012737	000002	000712	LT4ER2:	MOV	#2,EXFL	;FLAG RECVD: NOT EXPT
1931	004146	012737	024232	000616	LT4ERG:	MOV	#MSLT4,EMADDR	;SET LT4 HEADER
1932	004154	012737	021662	000666		MOV	#MSG9,ERADD	;SET ERROR CONDITION
1933	004162	012737	003716	000706		MOV	#LT4A,SCOLP	;SET SCOPE ADDRESS
1934	004170	004737	015230			JSR	PC,LTGER	;GO TO ERROR
1935	004174	000751				BR	LT4D	;IF NO SCOPE: BR
1936	004176	000137	002530		LT4X:	JMP	TSCD2	;RETURN TO SCHED
1937								


```

1978                                     ;LOGIC TEST 6: TC REGISTER BIT TEST*****
1979
1980 004372 000240                       LT6:  NOP
1981 004374 012737 024363 000616  LT6IT: MOV  #MSLT6,EMADDR ;POINT TO LT6 HEADER
1982 004402 012700 000003                MOV  #3,R0 ;SET NUMBER OF TESTS
1983 004406 005001                       LT6A1: CLR  R1
1984 004410 004737 016726                LT6A:  JSR  PC,INIT1 ;GO INIT
1985 004414 010177 174122                LT6B:  MOV  R1,@TC ;WRITE TC
1986 004420 017702 174116                MOV  @TC,R2 ;READ TC
1987 004424 042702 160000                BIC  #160000,R2 ;MASK OUT SAC
1988 004430 020102                       CMP  R1,R2 ;SEE IF EXPT = RECVD
1989 004432 001010                       BNE  LT6ER1 ;IF NOT: BR
1990 004434 005300                       LT6D:  DEC  R0
1991 004436 001417                       BEQ  LT6X ;IF DONE ALL: BR
1992 004440 022700 000001                CMP  #1,R0 ;SEE IF RESET TEST
1993 004444 001760                       BEQ  LT6A1 ;IF SO: BR
1994 004446 012701 017777                MOV  #17777,R1 ;SET TEST WORD
1995 004452 000756                       BR   LT6A ;DO SET TEST
1996 004454 012737 022036 000666  LT6ER1: MOV  #MSG18,ERADD ;SET ERROR CODE
1997 004462 012737 004414 000706        MOV  #LT6B,SCOLP ;SET SCOPE ADDRES
1998 004470 004737 016336                JSR  PC,LTGER1 ;GO TO ERROR
1999 004474 000757                       BR   LT6D ;CONTINUE
2000 004476 004737 016600                LT6X:  JSR  PC,ITER ;GO SEE IF ITERATIONS
2001 004502 000137 002530                JMP  TSCD2 ;RETURN TO SCHED
2002

```



```

2004                                     ;LOGIC TEST 7: FRAME COUNT BIT TEST*****
2005
2006 004506 000240                       LT7:  NOP
2007 004510 012700 000003                LT7IT: MOV  #3,R0          ;SET TEST NUMBER
2008 004514 012737 024432 000616        LT7C:  MOV  #MSLT7,EMADDR ;SET TEST HEADER
2009 004522 005001                       CLR  R1                ;SET TEST WORD
2010 004524 004737 016726                LT7A:  JSR  PC,INIT1    ;GO INIT
2011 004530 010177 173762                MOV  R1,@FC           ;CLEAR FRAME COUNT
2012 004534 017702 173756                MOV  @FC,R2          ;READ FC
2013 004540 020102                       CMP  R1,R2           ;SEE IF EXPT = RECVD
2014 004542 001010                       BNE  LT7ER1
2015 004544 005300                       LT7B:  DEC  R0                ;SEE IF DONE ALL
2016 004546 001417                       BEQ  LT7X            ;IF SO: BR
2017 004550 022700 000001                CMP  #1,R0           ;SEE IF RESET TEST
2018 004554 001757                       BEQ  LT7C            ;IF SO: BR
2019 004556 012701 177777                MOV  #-1,R1          ;SET TEST WORD TO -1
2020 004562 000760                       BR   LT7A            ;CONTINUE
2021 004564 012737 022055 000666        LT7ER1: MOV #MSG19,ERADD ;SET ERROR CODE
2022 004572 012737 004524 000706        MOV  #LT7A,SCOLP    ;SET SCOPE ADDRESS
2023 004600 004737 016336                JSR  PC,LTGER1      ;GO PRINT ERROR
2024 004604 000757                       BR   LT7B            ;ELSE CONTINUE
2025 004606 012700 000003                LT7X:  MOV  #3,R0          ;RESET TEST AMT
2026 004612 004737 016600                JSR  PC,ITER        ;GO SEE IF ITERATIONS
2027 004616 000137 002530                JMP  TSCD2          ;RETURN TO SCHED
2028

```

```

2030                                     ;LOGIC TEST 10: FUNCTION CODE BIT TEST*****
2031
2032 004622 000240                       LT10:  NOP
2033 004624 012737 024501 000616      LT10IT: MOV  #MSLT10,EMADDR ;SET TEST HEADER
2034 004632 012700 000003              MOV  #3,R0 ;SET NUMBER OF TESTS
2035 004636 005001                      LT10A1: CLR  R1 ;SET TEST WORRD
2036 004640 012777 000040 173652      LT10A:  MOV  #40,@CS ;INIT
2037 004646 013777 000620 173644      MOV  DRVN,@CS ;SELECT DRIVE
2038 004654 010177 173630              MOV  R1,@C1 ;WRITE C1
2039 004660 017702 173624              MOV  @C1,R2 ;READ C1
2040 004664 042702 177701              BIC  #177701,R2 ;MASK FUNCTION CODE
2041 004670 020102                      CMP  R1,R2 ;SEE IF EXPT = RECVD
2042 004672 001010                      BNE  LT10E1
2043 004674 005300                       LT10B:  DEC  R0
2044 004676 001417                      BEQ  LT10X ;IF DONE ALL: BR
2045 004700 022700 000001              CMP  #1,R0 ;SEE IF RESET TEST
2046 004704 001754                      BEQ  LT10A1 ;IF SO: BR
2047 004706 012701 000076              MOV  #76,R1 ;SET TEST WORD
2048 004712 000752                      BR   LT10A ;DO SET TEST
2049 004714 012737 022074 000666      LT10E1: MOV  #MSG20,ERADD ;SET ERROR CODE
2050 004722 012737 004640 000706      MOV  #LT10A,SCOLP ;SET SCOPE ADDRESS
2051 004730 004737 016336              JSR  PC,LTGER1 ;GO PRINT ERROR
2052 004734 000757                      BR   LT10B ;ELSE CONTINUE
2053 004736 004737 016600              LT10X: JSR  PC,ITER ;GO SEE IF ITERATIONS
2054 004742 000137 002530              JMP  TSCD2 ;RETURN TO SCHED

```



```

2056
2057
2058
2059 004746 000240          LT11:  NOP
2060 004750 012737 024557 000616  LT11IT: MOV    #MSLT11,EMADDR ;SET TEST HEADER
2061 004756 004737 016726          JSR    PC,INIT1    ;GO INIT
2062 004762 017702 173522          MOV    @C1,R2      ;READ C1
2063 004766 032702 000001          BIT    #1,R2       ;SEE IF GO=0
2064 004772 001030          BNE    LT11E1
2065 004774 012777 000015 173532  LT11B: MOV    #15,@MR    ;SELECT WAM 3
2066 005002 005077 173510          CLR    @FC         ;ASSURE FCS = 1
2067 005006 052777 001700 173526  BIS    #1700,@TC   ;ASSURE FMT OK
2068 005014 012777 000071 173466  MOV    #71,@C1     ;SET READ+GO
2069 005022 017702 173462          MOV    @C1,R2      ;READ C1
2070 005026 032702 000001          BIT    #1,R2       ;SEE IF GO =1
2071 005032 001424          BEQ    LT11E2
2072 005034 004737 016726          LT11C: JSR    PC,INIT1 ;GO INIT
2073 005040 017702 173444          MOV    @C1,R2      ;READ C1
2074 005044 032702 000001          BIT    #1,R2       ;SEE IF GO=0
2075 005050 001444          BEQ    LT11X
2076 005052 000430          BR     LT11E3
2077 005054 012737 022126 000666  LT11E1: MOV    #MSG21,ERADD ;SET ERROR CODE
2078 005062 012702 000001          MOV    #1,R2       ;SET REVD
2079 005066 005001          CLR    R1          ;SET EXPT
2080 005070 012737 004750 000706  MOV    #LT11IT,SCOLP ;SET SCOPE ADDRESS
2081 005076 004737 016336          JSR    PC,LTGER1   ;GO PRINT ERROR
2082 005102 000734          BR     LT11B
2083 005104 012737 022164 000666  LT11E2: MOV    #MSG22,ERADD ;SET ERROR CODE
2084 005112 005002          CLR    R2          ;SET RCVD
2085 005114 012701 000001          MOV    #1,R1       ;SET EXPT
2086 005120 012737 004774 000706  MOV    #LT11B,SCOLP ;SET SCOPE ADDRESS
2087 005126 004737 016336          JSR    PC,LTGER1   ;GO PRINT ERROR
2088 005132 000740          BR     LT11C
2089 005134 012737 022205 000666  LT11E3: MOV    #MSG23,ERADD ;SET ERROR CODE
2090 005142 005001          CLR    R1          ;SET EXPT
2091 005144 012702 000001          MOV    #1,R2       ;SET RCVD
2092 005150 012737 005034 000706  MOV    #LT11C,SCOLP ;SET SCOPE ADDRESS
2093 005156 004737 016336          JSR    PC,LTGER1   ;GO PRINT ERROR
2094 005162 004737 016600          LT11X: JSR    PC,ITER  ;GO SEE IF ITERATIONS
2095 005166 000137 002530          JMP    TSCD2       ;RETURN TO SCHED

```

```

2097
2098
2099
2100 005172 000240
2101 005174 012737 024624 000616
2102 005202 004737 016726
2103 005206 032777 000200 173306
2104 005214 001426
2105 005216 012777 000015 173310
2106 005224 005077 173266
2107 005230 052777 001700 173304
2108 005236 012777 000071 173244
2109 005244 032777 000200 173250
2110 005252 001020
2111 005254 004737 016726
2112 005260 032777 000200 173234
2113 005266 001033
2114 005270 000422
2115 005272 012737 022240 000666
2116 005300 012737 005174 000706
2117 005306 004737 016330
2118 005312 000741
2119 005314 012737 022266 000666
2120 005322 012737 005216 000706
2121 005330 004737 016330
2122 005334 000747
2123 005336 012737 022315 000666
2124 005344 012737 005254 000706
2125 005352 004737 016330
2126 005356 004737 016600
2127 005362 000137 002530

;LOGIC TEST 12: DRIVE READY BIT*****
LT12: NOP
LT12IT: MOV #MSLT12,EMADDR ;SET TEST HEADER
JSR PC,INIT1 ;GO INIT
BIT #200,@DS ;SEE IF DRY=1
BEQ LT12E1
LT12B: MOV #15,@MR ;SET WAM3
CLR @FC ;ASSURE FCS = 1
BIS #1700,@TC ;ASSURE FMT OK
MOV #71,@C1 ;SET READ*GO
BIT #200,@DS ;SEE IF DRY=0
BNE LT12E2
LT12C: JSR PC,INIT1 ;GO INIT
BIT #200,@DS ;SEE IF DRY=1
BNE LT12X ;IF SO: BR
BR LT12E3 ;ELSE GO TO ERROR 3
LT12E1: MOV #MSG24,ERADD ;SET ERROR CODE
MOV #LT12IT,SCOLP ;SET SCOPE ADDRESS
JSR PC,LTGER2 ;GO TO ERROR
BR LT12B ;CONTINUE
LT12E2: MOV #MSG25,ERADD ;SET ERROR CODE
MOV #LT12B,SCOLP ;SET LOOP ADDRESS
JSR PC,LTGER2 ;GO PRINT ERROR
BR LT12C ;CONTINUE
LT12E3: MOV #MSG25A,ERADD ;SET ERROR CODE
MOV #LT12C,SCOLP ;SET ERROR LOOP
JSR PC,LTGER2 ;GET PRINT ERROR
LT12X: JSR PC,ITER ;GO TO ITERATION SUBROUTINE
JMP TSCD2 ;RETURN TO SCHED

```



```

2129
2130
2131
2132 005366 005000
2133 005370 012737 024675 000616
2134 005376 004737 016726
2135 005402 012737 005460 000664
2136 005410 005077 173074
2137 005414 005077 173146
2138 005420 052777 000100 173062
2139 005426 005300
2140 005430 001376
2141 005432 012777 000340 173126
2142 005440 012737 022342 000666
2143 005446 012737 005376 000706
2144 005454 004737 016330
2145 005460 004737 016600
2146 005464 000137 002530

;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 CS1
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

```

05

2148
 2149
 2150
 2151
 2152
 2153
 2154
 2155
 2156
 2157
 2158
 2159
 2160
 2161
 2162
 2163
 2164
 2165
 2166
 2167
 2168
 2169
 2170
 2171
 2172
 2173

;THE NEXT 4 TESTS ARE MANUAL INTERVENTION STATUS TESTS.
 ;THE OPERATOR WILL BE REQUIRED TO MANIPULATE THE TE16
 ;CONTROL PANEL IN ACCORDANCE WITH TTY INSTRUCTIONS.

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

2155	005470	032777	001000	173072	LT14:	BIT	#1000,@SWR	;SEE IF INHIB MAN TST
2156	005476	001005				BNE	LT14A	;IF NOT: BR
2157	005500	005737	000734			TST	STFLG	;SEE IF SINGLE TEST
2158	005504	001433				BEQ	LT14XX	;IF NOT: BR
2159	005506	000137	016650			JMP	INMT	;ELSE GO PRINT INHIB MSG
2160	005512	012737	024742	000616	LT14A:	MOV	#MSLT14,EMADDR	;SET TEST HEADER
2161	005520	012704	027147			MOV	#MMSG1,R4	;SET INSTRUCTION ONE
2162	005524	004737	017200			JSR	PC,INST	;GO DO INSTRUCTION
2163	005530	004737	016726		LT14IT:	JSR	PC,INIT1	;INIT, SELECT DRIVE * SLAVE
2164	005534	012701	014602			MOV	#14602,R1	;SET TEST WORD
2165	005540	017702	172756			MOV	@DS,R2	;ASSURE MOL,WRL,DPR,DRY,BOT
2166	005544	020102				CMP	R1,R2	
2167	005546	001410				BEQ	LT14X	;IF SO: BR
2168	005550	012737	005530	000706		MOV	#LT14IT,SCOLP	;SET LOOP ADDRESS
2169	005556	012737	022371	000666		MOV	#MSG27,ERADD	;SET ERROR CODE
2170	005564	004737	016336			JSR	PC,LTGER1	;GO PRINT ERROR
2171	005570	004737	016600		LT14X:	JSR	PC,ITER	;GO SEE IF ITERATION
2172	005574	000137	002530		LT14XX:	JMP	TSCD2	;RETURN TO SCHED


```

2175
2176
2177
2178 005600 032777 001000 172762 LT15: BIT #1000,@SWR ;SEE IF INHIB MAN TST
2179 005606 001005 BNE LT15A ;IF NOT: BR
2180 005610 005737 000734 TST STFLG ;SEE IF SINGLE TEST
2181 005614 001433 BEQ LT15XX ;IF NOT: BR
2182 005616 000137 016650 JMP INMT ;ELSE GO PRINT INHIB MSG
2183 005622 012737 025051 000616 LT15A: MOV #MSLT15,EMADDR ;SET TEST HEADER
2184 005630 012704 027245 MOV #MMSG2,R4
2185 005634 004737 017200 JSR PC,INST ;PRINT INSTRUCTION
2186 005640 004737 016736 LT15IT: JSR PC,INIT2 ;GO INIT, SELECT DRIVE, SLAV
2187 005644 012701 100700 MOV #100700,R1 ;SET TEST WORD
2188 005650 017702 172646 MOV @DS,R2 ;READ STATUS
2189 005654 020102 CMP R1,R2 ;SEE OF EXPT-RCVD
2190 005656 001410 BEQ LT15X
2191 005660 012737 005640 000706 MOV #LT15IT,SCOLP ;SET LOOP ADDRESS
2192 005666 012737 022371 000666 MOV #MSG27,ERADD ;SET ERROR CODE
2193 005674 004737 016336 JSR PC,LTGER1 ;GO PRINT ERROR
2194 005700 004737 016600 LT15X: JSR PC,ITER ;GO SEE IF ITERATIONS
2195 005704 000137 002530 LT15XX: JMP TSCD2 ;RETURN TO SCHED

```

```

2197
2198
2199
2200 005710 032777 001000 172652 LT16: BIT #1000,@SWR ;SEE IF INHIB MAN TST
2201 005716 001005 BNE LT16A ;IF NOT: BR
2202 005720 005737 000734 TST STFLG ;SEE IF SINGLE TEST
2203 005724 001433 BEQ LT16XX ;IF NOT: BR
2204 005726 000137 016650 JMP INMT ;ELSE GO PRINT INHIB MSG
2205 005732 012737 025141 000616 LT16A: MOV #MSLT16,EMADDR ;SET TEST HEADER
2206 005740 012704 027266 MOV #MMSG3,R4
2207 005744 004737 017200 JSR PC,INST ;GO PRINT INSTRUCTION
2208 005750 004737 016736 LT16IT: JSR PC,INIT2 ;SELECT DRIVE,SLAVE
2209 005754 013701 006020 MOV STWD16,R1 ;SET TEST WORD (ATA!MOL!WRL!EOT!DPR!DR!SSC!SLA)
2210 005760 017702 172536 MOV @DS,R2 ;READ STATUS
2211 005764 020102 CMP R1,R2 ;SEE IF EXPT=RCVD
2212 005766 001410 BEQ LT16X ;IF SO: BR
2213 005770 012737 005750 000706 MOV #LT16IT,SCOLP ;SET LOOP ADDRESS
2214 005776 012737 022371 000666 MOV #MSG27,ERADD ;SET ERROR CODE
2215 006004 004737 016336 JSR PC,LTGER1 ;GO PRINT ERROR
2216 006010 004737 016600 LT16X: JSR PC,ITER ;GO SEE IF ITERATION
2217 006014 000137 002530 LT16XX: JMP TSCD2 ;RETURN TO SCHED
2218 006020 116701 STWD16: .WORD 116701 ;SET UP TEST WORD FOR TE16
2219 ;CONTENTS OF TEST WORD FOR
2220 ;A TU77 IS 116741
2221

```



```

2223
2224                                     ;LOGIC TEST 17: STATUS AT ON LINE, LOADED*****
2225
2226 006022 032777 001000 172540 LT17: BIT #1000,@SWR ;SEE IF INHIB MAN TST
2227 006030 001005 BNE LT17A ;IF NOT: BR
2228 006032 005737 000734 TST STFLG ;SEE IF SINGLE TEST
2229 006036 001433 BEQ LT17XX ;IF NOT: BR
2230 006040 000137 016650 JMP INMT ;ELSE GO PRINT INHIB MSG
2231 006044 012737 025230 000616 LT17A: MOV #MSLT17,EMADDR ;SET TEST HEADER
2232 006052 012704 027324 MOV #MMSG4,R4
2233 006056 004737 017200 JSR PC,INST ;GO PRINT INSTRUCTION
2234 006062 004737 016736 LT17IT: JSR PC,INIT2 ;SELECT DRIVE, SLAVE
2235 006066 013701 006132 MOV STWD17,R1 ;SET TEST WORD
2236 006072 017702 172424 MOV @DS,R2 ;READ STATUS
2237 006076 020102 CMP R1,R2 ;SEE IF EXPT=RCVD
2238 006100 001410 BEQ LT17X ;IF SO: BR
2239 006102 012737 006062 000706 MOV #LT17IT,SCOLP ;SET LOOP ADDRESS
2240 006110 012737 022371 000666 MOV #MSG27,ERADD ;SET ERROR CODE
2241 006116 004737 016336 JSR PC,LTGER1 ;YES PRINT ERROR
2242 006122 004737 016600 LT17X: JSR PC,ITER ;GO SEE IF ITERATIONS
2243 006126 000137 002530 LT17XX: JMP TSCD2 ;RETURN TO SCHED
2244 006132 110701 STWD17: .WORD 110701 ;TEST WORD FOR TE16
2245 ;CONTENTS OF TEST WORD
2246 ;IS 110741 FOR A TU77

```

2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280

;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)*****

```

2256 006134 012737 025311 000616 LT20:  MOV    #MSLT20,EMADDR ;SET TEST HEADER
2257 006142 012737 006162 000706      MOV    #LT20A,SCOLP  ;SET LOOP ADDRESS
2258 006150 012700 000022      LT20IT: MOV    #22,R0      ;SET NUMBER OF ILL CODES
2259 006154 012737 000544 000670      MOV    #ILFT,TEMP1  ;POINT TO START IF TABLE
2260 006162 004737 016726      LT20A:  JSR    PC,INIT1 ;GO INIT, SELECT SLAVE * DRIVE
2261 006166 012777 177777 172316      MOV    #-1,@WC      ;SET WC= -1
2262 006174 012701 000001      MOV    #1,R1        ;SET TEST WORD
2263 006200 117777 172464 172302      MOVB   @TEMP1,@C1   ;SET ILL CODE
2264 006206 017702 172312      MOV    @ER,R2       ;READ ER
2265 006212 030102      BIT    R1,R2        ;SEE IF EXPT=RCVD
2266 006214 001011      BNE    LT20B        ;IF SO: BR
2267 006216 012737 027653 000666      MOV    #TMS17,ERADD ;SET ERROR CODE
2268 006224 012737 000001 000712      MOV    #1,EXFL      ;SET EXPT FLG
2269 006232 004737 015222      JSR    PC,LTGERO    ;GO PRINT ERROR
2270 006236 000404      BR     LT20C
2271 006240 020102      LT20B:  CMP    R1,R2        ;SEE UNEXPECTED ERRORS
2272 006242 001402      BEQ    LT20C        ;IF NOT: BR
2273 006244 004737 015210      JSR    PC,LTGER3    ;ELSE PRINT ERROR
2274 006250 005300      LT20C:  DEC    R0           ;SEE IF DONE ALL ILL CODES
2275 006252 001403      BEQ    LT20X        ;IF SO: BR
2276 006254 005237 000670      INC    TEMP1        ;BUMP ADDRESS
2277 006260 000740      BR     LT20A        ;CONTINUE
2278 006262 004737 016600      LT20X:  JSR    PC,ITER     ;GO SEE IF ITERATION
2279 006266 004737 015670      JSR    PC,DRVCLR    ;
2280 006272 000137 002530      JMP    TSCD2        ;RETURN TO SCHED
    
```



```

2282
2283           ;LOGIC TEST 21: REGISTER MODIFICATION REFUSED(RMR)*****
2284
2285 006276 012737 025370 000616 LT21:  MOV    #MSLT21,EMADDR ;SET TEST HEADER
2286 006304 012737 006312 000706      MOV    #LT21IT,SCOLP ;SET SCOPE LOOP ADDRESS
2287 006312 004737 016726      LT21IT: JSR   PC,INIT1 ;GO INIT, SELECT SLAVE, DRIVE.
2288 006316 052777 000300 172216      BIS    #300,@TC ;SET FORMAT
2289 006324 012777 000015 172202      MOV    #15,@MR ;SET WAM3
2290 006332 012777 000071 172150      MOV    #71,@C1 ;SET READ.GO
2291 006340 005077 172152      CLR    @FC ;ATTEMPT WRITE TO FC
2292 006344 012701 000004      MOV    #4,R1 ;SET TEST WORD
2293 006350 017702 172150      MOV    @ER,R2 ;GET ER
2294 006354 030102      BIT    R1,R2 ;SEE IF EXPT-RCVD
2295 006356 001011      BNE    LT21A ;IF SO: BR
2296 006360 012737 027667 000666      MOV    #TMS19,ERADD ;SET ERROR CODE
2297 006366 012737 000001 000712      MOV    #1,EXFL ;SET EXPT FLG
2298 006374 004737 015222      JSR   PC,LTGERO ;GO PRINT ERROR
2299 006400 000404      BR     LT21B
2300 006402 020102      LT21A: CMP    R1,R2 ;SEE IF UNEXPECTED ERRORS
2301 006404 001402      BEQ    LT21B ;IF NOT: BR
2302 006406 004737 015210      JSR   PC,LTGER3 ;ELSE GO PRINT ERROR
2303           ;..B LT21B: JSR   PC,ITER ;..B DELETED GO SEE IF ITERATION
2304 006412 012703 040000      LT21B: MOV    #40000,R3
2305 006416 005303      LT21XA: DEC   R3 ;DELAY FOR ALPHA
2306 006420 001376      BNE    LT21XA
2307 006422 004737 015054      JSR   PC,EORPA ;GO DO EOR CLEAR
2308 006426 004737 015670      JSR   PC,DRVCLR
2309 006432 004737 016600      JSR   PC,ITER ;..B GO SEE IF ITERATION
2310 006436 000137 002530      JMP    TSCD2 ;RETURN TO SCHED

```

```

2312
2313
2314
2315 006442 012737 025424 000616 LT22: MOV #MSLT22,EMADDR ;SET TEST HEADER
2316 006450 012737 006456 000706 MOV #LT22IT,SCOLP ;SET SCOPE LOOP ADDRESS
2317 006456 004737 016726 LT22IT: JSR PC,INIT1 ;INIT, SELECT SLAVE DRIVE
2318 006462 052777 000020 172030 BIS #20,@CS ;ENABLE EVEN PARITY ON MB
2319 006470 012777 177777 172020 MOV #-1,@FC ;WRITE TO FC
2320 006476 012701 000010 MOV #10,R1 ;SET TEST WORD
2321 006502 042777 000020 172010 BIC #20,@CS ;RESET PARITY TO ODD
2322 006510 017702 172010 MOV @ER,R2 ;GET ER
2323 006514 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2324 006516 001011 BNE LT22A ;IF SO: BR
2325 006520 012737 027675 000666 MOV #TMS20,ERADD ;SET ERROR CODE
2326 006526 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2327 006534 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2328 006540 000404 BR LT22X
2329 006542 020102 LT22A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2330 006544 001402 BEQ LT22X ;IF NOT: BR
2331 006546 004737 015210 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2332 006552 004737 016600 LT22X: JSR PC,ITER ;GO SEE IF ITERATION
2333 006556 004737 015670 JSR PC,DRVCLR
2334 006562 000137 002530 JMP TSCD2 ;RETURN TO SCHED

```



```

2336
2337
2338
2339 006566 012737 025461 000616 LT23: MOV #MSLT23,EMADDR ;SET TEST HEADER
2340 006574 012737 006602 000706 MOV #LT23IT,SCOLP ;SET SCOPE ADDRESS
2341 006602 004737 016726 LT23IT: JSR PC,INIT1 ;GO INIT SELECT DRIVE+SLAVE
2342 ;**B BIC #360,@TC ;**B DELETED SET ILLEGAL FORMAT
2343 006606 052777 000360 171726 BIS #360,@TC ;**B SET ILLEGAL FORMAT FOR BOTH M8906 & M8915
2344 006614 012701 000020 MOV #20,R1 ;SET TEST WORD
2345 006620 012777 000015 171706 MOV #15,@MR ;SET WAM 3
2346 006626 012777 000071 171654 MOV #71,@C1 ;SET READ+GO
2347 006634 017702 171664 MOV @ER,R2 ;READ ER
2348 006640 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2349 006642 001011 BNE LT23A ;IF SO: BR
2350 006644 012737 027704 000666 MOV #TMS21,ERADD ;SET ERROR CODE
2351 006652 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2352 006660 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2353 006664 000404 BR LT23X
2354 006666 020102 LT23A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2355 006670 001402 BEQ LT23X ;IF NOT: BR
2356 006672 004737 015210 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2357 006676 004737 016600 LT23X: JSR PC,ITER ;GO SEE IF ITERATION
2358 006702 004737 015054 JSR PC,EORPA
2359 006706 004737 015670 JSR PC,DRVCLR
2360 006712 000137 002530 JMP TSCD2 ;RETURN TO SCHED

```

```

2362 ;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)*****
2363
2364 006716 012737 025526 000616 LT24: MOV #MSLT24,EMADDR ;SET TEST HEADER
2365 006724 012737 006732 000706 MOV #LT24IT,SCOLP ;SET SCOPE ADDRESS
2366 006732 012737 000005 000602 LT24IT: MOV #5,ITAMT
2367 006740 004737 016754 JSR PC,INIT3 ;GO INIT, SELECT DRIVE+SLAVE
2368 006744 052777 000300 171570 BIS #300,@TC ;SET NORMAL FORMAT
2369 006752 012777 030204 171534 MOV #WDATA,@BA ;SET BA
2370 006760 012777 177760 171530 MOV #-20,@FC ;SET FC
2371 006766 012777 177770 171516 MOV #-10,@WC ;SET WC
2372 006774 012777 000013 171532 MOV #13,@MR ;SELECT WAM 2
2373 007002 012777 000061 171500 MOV #61,@C1 ;SET WRITE+GO
2374 007010 052777 000020 171502 BIS #20,@CS ;FORCE EVEN PARITY
2375 007016 012701 000040 MOV #40,R1 ;SET TEST WORD
2376 007022 012703 000004 MOV #4,R3
2377 007026 005000 CLR R0
2378 007030 005300 1$: DEC R0
2379 007032 001376 BNE 1$ ;DELAY
2380 007034 005303 DEC R3
2381 007036 001374 BNE 1$
2382 007040 012700 000004 MOV #4,R0
2383 007044 012777 000013 171462 LT24B: MOV #13,@MR ;CLOCK MR 4 TIMES
2384 007052 005300 DEC R0
2385 007054 022700 000002 CMP #2,R0 ;SEE IF DONE 1 BYTE
2386 007060 001002 BNE LT24B0 ;IF NOT: BR
2387 007062 017701 171446 MOV @MR,R1 ;ELSE GET BYTE 1
2388 007066 005700 LT24B0: TST R0 ;SEE IF BYTE 2
2389 007070 001365 BNE LT24B ;IF NOT: BR
2390 007072 017704 171436 MOV @MR,R4 ;GET BYTE 2
2391 007076 005000 CLR R0
2392 007100 005300 LT24C: DEC R0
2393 007102 001376 BNE LT24C ;DELAY
2394 007104 032777 000040 171412 BIT #40,@ER ;SEE IF DPAR IS SET
2395 007112 001023 BNE LT24D ;IF SO: BR
2396 007114 000301 SWAB R1
2397 007116 042701 177400 BIC #177400,R1 ;GET LOW BYTE
2398 007122 042704 000377 BIC #377,R4
2399 007126 050401 BIS R4,R1 ;GET HIGH BYTE
2400 007130 005237 000740 INC T24FL ;SET T24 FLAG
2401 007134 012737 027712 000666 MOV #TMS22,ERADD ;SET ERROR CODE
2402 007142 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2403 007150 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2404 007154 005037 000740 CLR T24FL ;CLEAR FLAG
2405 007160 000412 BR LT24X
2406 007162 012701 000050 LT24D: MOV #50,R1
2407 007166 017702 171332 MOV @ER,R2 ;GET ERROR REGISTER
2408 007172 042702 020000 BIC #20000,R2 ;MASK OPI
2409 007176 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2410 007200 001402 BEQ LT24X ;IF NOT: BR
2411 007202 004737 015210 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2412 007206 042777 000020 171304 LT24X: BIC #20,@CS ;RESET EVEN PARITY
2413 007214 004737 015054 JSR PC,EORPA ;GO DO EOR CLEAR
2414 007220 004737 015670 JSR PC,DRVCLR ;GO SEE IF DRIVE CLEAR OK
2415 007224 004737 016600 JSR PC,ITER ;GO SEE IF ITERATION
2416 007230 012737 000020 000602 MOV #20,ITAMT
2417 007236 000137 002530 JMP TSCD2 ;RETURN TO SCHED

```



```

2419
2420           ;LOGIC TEST 25: NON-EXECUTABLE FUNCTION(NEF)*****
2421
2422 007242 012737 025566 000616 LT25:  MOV    #MSLT25,EMADDR ;SET TEST HEADER
2423 007250 004737 016754          LT25IT: JSR    PC,INIT3    ;INIT, SELECT DRIVE+SLAVE
2424 007254 052777 000300 171260          BIS    #300,@TC    ;SET NORMAL FORMAT
2425 007262 012777 177777 171226          MOV    #-1,@FC    ;SET ITLLEGAL FC
2426 007270 012777 000013 171236          MOV    #13,@MR    ;SET WAM 2
2427 007276 012777 000061 171204          MOV    #61,@C1    ;LOAD WRITE+GO
2428 007304 012701 004000          MOV    #4000,R1   ;SET TEST WORD
2429 007310 017702 171210          MOV    @ER,R2    ;GET ER
2430 007314 030102          BIT    R1,R2    ;SEE IF EXPT=RCVD
2431 007316 001014          BNE    LT25A    ;IF SO: BR
2432 007320 012737 007250 000706          MOV    #LT25IT,SCOLP ;SET LOOP ADDRESS
2433 007326 012737 030000 000666          MOV    #TMS31,ERADD ;SET ERROR CODE
2434 007334 012737 000001 000712          MOV    #1,EXFL   ;SET EXPT FLAG
2435 007342 004737 015222          JSR    PC,LTGERO ;GO PRINT ERROR
2436 007346 000404          BR    LT25X
2437 007350 020102          LT25A: CMP    R1,R2    ;SEE IF UNEXPECTED ERRORS
2438 007352 001402          BEQ    LT25X    ;IF NOT: BR
2439 007354 004737 015210          JSR    PC,LTGER3 ;ELSE GO PRINT ERROR
2440 007360 004737 016600          LT25X: JSR    PC,ITER  ;GO SEE IF ITERATION
2441 007364 004737 015670          JSR    PC,DRVCLR
2442 007370 000137 002530          JMP    TSCD2    ;RETURN TO SCHED

```

```

2444
2445                                     ;LOGIC TEST 26: FRAME COUNT ERROR(FCE)*****
2446
2447 007374 012737 025622 000616 LT26: MOV #MSLT26,EMADDR ;SET TEST HEADER
2448 007402 004737 016754 LT26IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2449 007406 005000 CLR R0
2450 007410 005300 1$: DEC R0
2451 007412 001376 BNE 1$ ;AWAIT OPI RESET
2452 007414 052777 000300 171120 BIS #300,@TC ;SET NORMAL FORMAT
2453 007422 012777 177770 171062 MOV #-10,@WC ;SET WC=-10
2454 007430 012777 177760 171060 MOV #-20,@FC ;SET FC=-20
2455 007436 012777 000013 171070 MOV #13,@MR ;SET WAM 3
2456 007444 012777 000061 171036 MOV #61,@C1 ;LOAD WRITE+GO
2457 007452 012701 001000 MOV #1000,R1 ;SET TEST WORD
2458 007456 005000 CLR R0
2459 007460 005300 2$: DEC R0
2460 007462 001376 BNE 2$ ;DELAY
2461 007464 012777 000025 171042 MOV #25,@MR ;LOAD MM EOR CLEAR
2462 007472 105077 171036 CLR R0 ;RESET MR
2463 007476 012703 000004 MOV #4,R3
2464 007502 005000 CLR R0
2465 007504 032777 001000 171012 3$: BIT #1000,@ER ;SEE IF FCE SET
2466 007512 001022 BNE 4$ ;IF SO: BR
2467 007514 005300 DEC R0
2468 007516 001372 BNE 3$ ;DELAY
2469 007520 005303 DEC R3
2470 007522 001370 BNE 3$
2471 007524 017702 170774 MOV @ER,R2 ;GET ER
2472 007530 012737 007402 000706 MOV #LT26IT,SCOLP ;SET SCOPE ADDRESS
2473 007536 012737 027757 000666 MOV #TMS28,ERADD
2474 007544 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2475 007552 004737 015222 JSR PC,LTGER0 ;GO PRINT ERROR
2476 007556 000406 BR LT26X
2477 007560 017702 170740 4$: MOV @ER,R2 ;GET ERROR REGISTER
2478 007564 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2479 007566 001402 BEQ LT26X ;IF NOT: BR
2480 007570 004737 015210 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2481 007574 004737 016600 LT26X: JSR PC,ITER ;GO SEE IF ITERATION
2482 007600 004737 015670 JSR PC,DRVCLR
2483 007604 000137 002530 JMP TSCD2 ;RETURN TO SCHED

```



```

2485
2486 ;LOGIC TEST 27: ILLEGAL REGISTER(ILR)*****
2487
2488 007610 022737 172400 000510 LT27:  CMP #172400,C1 ;SEE IF ADDRESSES OPEN
2489 007616 001041 ;IF NOT: BR
2490 007620 012737 007644 000706 ;SET SCOPE ADDRESS
2491 007626 012737 025656 000616 ;SET TEST HEADER
2492 007634 012700 000020 ;SET NUMBER OF ILR TESTS
2493 007640 012701 172434 ;SET FIRST ILR ADDRESS
2494 007644 004737 016754 LT27A: JSR PC,INIT5 ;GO INIT, SELECT DRIVE+SLAVE
2495 007650 011103 ;ATTEMPT ILR READ
2496 007652 032777 000002 170644 ;SEE IF ILR=1
2497 007660 001010 ;IF SO: BR
2498 007662 012737 000001 000712 ;SET EXPT-NOT RCVD FLAG
2499 007670 012737 027601 000666 ;SET ERROR CODE
2500 007676 004737 015230 ;GO PRINT ERROR
2501 007702 005300 LT27B: DEC R0 ;SEE IF DONE ALL
2502 007704 001402 ;IF SO: BR
2503 007706 005721 ;BUMP ADDRESS
2504 007710 000755 BR LT27A ;CONTINUE TESTS
2505 007712 004737 016600 LT27X: JSR PC,ITER ;GO SEE IF ITERATIONS
2506 007716 004737 015670 JSR PC,DRVCLR
2507 007722 000137 002530 LT27XX: JMP TSCD2 ;RETURN TO SCHED

```

```

2509
2510
2511
2512 007726 012737 030006 000666 LT30: MOV @TMS32,ERADD ;SET ERROR CODE
2513 007734 012737 025712 000616 MOV @MSLT30,EMADDR ;SET TEST HEADER
2514 007742 012737 007750 000706 MOV @LT30IT,SCOLP ;SET SCOPE ADDRESS
2515 007750 004737 016754 LT30IT: JSR PC,INIT3 ;INIT, SELECT DRIVE * SLAVE
2516 007754 052777 000300 170560 BIS @300,@TC ;SET NORMAL FORMAT
2517 007762 012701 010000 MOV @10000,R1 ;SET TEST WORD
2518 007766 012777 000017 170540 MOV @17,@MR ;CRIPPLE OCCUPIED
2519 007774 005077 170516 CLR @FC ;SET FC3
2520 010000 012777 000061 170502 MOV @61,@C1 ;LOAD WRITE*GO
2521 010006 032777 010000 170510 BIT @10000,@ER ;SEE IF DTE SET
2522 010014 001005 BNE LT30A ;IF SO: BR
2523 010016 012737 000001 000712 MOV @1,EXFL ;SET EXPT FLG
2524 010024 004737 015222 JSR PC,LTGERO ;GO PRINT ERRGR
2525 010030 004737 016754 LT30A: JSR PC,INIT3 ;GO INIT SELECT DRIVE,SLAVE
2526 010034 052777 000300 170500 BIS @300,@TC ;SET FORMAT
2527 010042 012701 010000 MOV @10000,R1 ;SET TEST WORD
2528 010046 005077 170444 CLR @FC ;SET FCS
2529 010052 012777 000015 170454 MOV @15,@MR ;SET WRAP 3
2530 010060 012777 000061 170422 MOV @61,@C1 ;LOAD WRITE*GO
2531 010066 012704 040000 MOV @40000,R4
2532 010072 005777 170444 LT30B: TST @TC ;SEE IF ALPHA
2533 010076 100015 BPL LT30C ;AWAIT ALPHA
2534 010100 005300 DEC R0
2535 010102 001373 BNE LT30B
2536 010104 013704 000616 MOV EMADDR,R4
2537 010110 004737 017760 JSR PC,TTOUT ;PRINT HEADER
2538 010114 012704 023125 MOV @MSG50,R4
2539 010120 004737 017760 JSR PC,TTOUT ;PRINT ALPHA ERROR
2540 010124 004737 016550 JSR PC,SCOPE
2541 010130 000435 BR LT30X
2542 010132 012777 000015 170374 LT30C: MOV @15,@MR ;CLOCK MR
2543 010140 012777 000015 170366 MOV @15,@MR ;CLOCK MR
2544 010146 005000 CLR R0
2545 010150 005300 LT30D: DEC R0
2546 010152 001376 BNE LT30D ;DELAY
2547 010154 032777 010000 170342 BIT @10000,@ER ;SEE IF DTE SET
2548 010162 001006 BNE LT30E ;IF SO: BR
2549 010164 012737 000001 000712 MOV @1,EXFL ;SET EXPT FLG
2550 010172 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2551 010176 000412 BR LT30X
2552 010200 012701 010000 LT30E: MOV @10000,R1 ;SET TEST WORD
2553 010204 017702 170314 MOV @ER,R2 ;GET ERROR REGISTER
2554 010210 042702 020100 BIC @20100,R2 ;MASK OPI AND VPE
2555 010214 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2556 010216 001402 BEQ LT30X ;IF NOT: BR
2557 010220 004737 015210 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2558 010224 004737 016600 LT30X: JSR PC,ITER ;GO SEE IF ITERATION
2559 010230 004737 015054 JSR PC,EORPA ;GO CLEAR GO BIT
2560 010234 004737 015670 JSR PC,DRVCLR
2561 010240 000137 002530 JMP TSCD2 ;RETURN TO SCHED
2562

```



```

2564
2565          ;LOGIC TEST 31: OPERATION INCOMPLETE(OPI)*****
2566
2567 010244 012737 025750 000616 LT31:  MOV    @MSLT31,EMADDR ;SET TEST HEADER
2568 010252 012737 010252 000706 LT31IT: MOV    @LT31IT,SCOLP ;SET SCOPE ADDRESS
2569 010260 012737 030022 000666      MOV    @TMS33A,ERADD ;SET ERROR MSG HDR
2570 010266 012737 000002 000602      MOV    @2,ITAMT ;SET REDUCED ITER COUNT
2571 010274 004737 016754      JSR    PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2572 010300 005000      CLR    R0
2573 010302 005300      1$:   DEC    R0
2574 010304 001376      BNE    1$ ;AWAIT OPI RESET
2575 010306 052777 000300 170226      BIS    @300,@TC ;SET FORMAT
2576 010314 012777 000013 170212      MOV    @13,@MR ;SET WAM 2
2577 010322 005077 170170      CLR    @FC ;SET FRAME COUNT
2578 010326 012705 020000      MOV    @20000,R5 ;SET TEST BIT (OPI)
2579 010332 012702 010350      MOV    @2$,R2 ;SET RETURN ADDRESS FROM TIMER
2580 010336 004737 010550      JSR    PC,TIMON ;START TIMER
2581 010342 012777 000061 170140      MOV    @61,@C1 ;LOAD WRITE+GO
2582 010350 030577 170150      2$:   BIT    R5,@ER ;BRANCH WHEN OPI SETS
2583 010354 001002      BNE    3$
2584 010356 000163 010642      JMP    TIMER(R3) ;GO TO TIMER & RETURN TO 2$ ABOVE
2585 010362 017702 170136      3$:   MOV    @ER,R2 ;GET ERROR REGISTER
2586 010366 020502      CMP    R5,R2 ;SEE IF UNEXPECTED ERRORS
2587 010370 001403      BEQ    4$ ;IF NOT: BR
2588 010372 004737 015210      JSR    PC,LTGER3 ;ELSE PRINT ERROR
2589 010376 000453      BR     LT31X
2590 010400 004737 010734      4$:   JSR    PC,TIMOK ;GO CHECK TIME FOR OPI TO SET
2591 010404 102450      BVS    LT31X ;BRANCH IF TIME WAS INCORRECT
2592
2593 010406 012737 010422 000706      MOV    @LT31A,SCOLP ;SET SCOPE LOOP
2594 010414 012737 030036 000666      MOV    @TMS33B,ERADD ;SET ERROR MSG HEADER
2595 010422 004737 016754      LT31A: JSR    PC,INIT3 ;GO INIT
2596 010426 005000      CLR    R0
2597 010430 005300      1$:   DEC    R0 ;WAIT FOR OPI TO CLEAR
2598 010432 001376      BNE    1$
2599 010434 052777 000300 170100      BIS    @300,@TC ;SET FORMAT
2600 010442 012777 000015 170064      MOV    @15,@MR ;SET WRAP 3
2601 010450 012702 010472      MOV    @2$,R2 ;SET RETURN ADDRESS FROM TIMER
2602 010454 012705 020000      MOV    @20000,R5 ;SET TEST WORD
2603 010460 004737 010550      JSR    PC,TIMON ;START TIMER
2604 010464 012777 000071 170016      MOV    @71,@C1 ;LOAD READ+GO
2605 010472 030577 170026      2$:   BIT    R5,@ER ;BRANCH WHEN OPI SETS
2606 010476 001002      BNE    3$
2607 010500 000163 010642      JMP    TIMER(R3) ;GO TO TIMER
2608 010504 017702 170014      3$:   MOV    @ER,R2 ;GET ERROR REGISTER
2609 010510 020502      CMP    R5,R2 ;SEE IF UNEXPECTED ERRORS
2610 010512 001403      BEQ    4$ ;ELSE PRINT ERROR
2611 010514 004737 015210      JSR    PC,LTGER3 ;EXIT TEST
2612 010520 000402      BR     LT31X
2613 010522 004737 010734      4$:   JSR    PC,TIMOK ;GO CHECK TIME
2614 010526 004737 016600      LT31X: JSR    PC,ITER ;GO SEE IF ITERATIONS
2615 010532 004737 015670      JSR    PC,DRVCLR
2616 010536 012737 000020 000602      MOV    @20,ITAMT
2617 010544 000137 002530      JMP    TSCD2 ;RETURN TO SCHED
2618
2619          ;ROUTINE TO START THE TIMER, THE TIMER IS AN OSCILLATOR IN THE MAIN

```

```

2620 ;ENANCE REGISTER (BIT 6) THAT TOGGLES EVERY 56 (10) MICROSECONDS. THIS
2621 ;ROUTINE WAITS FOR THE OSCILLATOR TO TOGGLE AND RETURN WITH R3 INDICATING
2622 ;THE STATE OF THE OSCILLATOR.
2623 010550 005000 TIMON: CLR R0 ;CLEAR TICK COUNT
2624 010552 005001 CLR R1
2625 010554 012703 000024 MOV #24,R3 ;PRESET INDEX TO TIMER
2626 010560 032777 000100 167746 BIT #100,@MR ;BRANCH IF OSC CLEAR
2627 010566 001405 BEQ 2$
2628 010570 032777 000100 167736 1$: BIT #100,@MR ;WAIT FOR OSC TO CLEAR
2629 010576 001374 BNE 1$
2630 010600 000405 BR 4$ ;EXIT
2631
2632 010602 005403 2$: NEG R3 ;SET INDEX TO TIMER
2633 010604 032777 000100 167722 3$: BIT #100,@MR ;WAIT FOR OSC TO SET
2634 010612 001774 BEQ 3$
2635 010614 000207 4$: RTS PC ;RETURN
2636
2637 ;THIS ROUTINE TIMES AN EVENT. EACH TIME THE OSCILLATOR BIT CHANGES
2638 ;STATE THE TICK COUNT IN R1 & R0 IS INCREMENTED. THE ROUTINE IS CALLED
2639 ;USING R3 AS AN INDEX TO INDICATE THE OSCILLATORS PAST STATE. WHEN
2640 ;THE OSC BIT CHANGES STATE R3 IS NEGATED.
2641 010616 032777 000100 167710 TIMER1: BIT #100,@MR ;BRANCH IF OSC HAS CHANGED STATE
2642 010624 001406 BEQ TIMER
2643 010626 000112 JMP (R2) ;RETURN
2644 010642 005403 .-TIMER!,24
2645 010644 062700 000001 TIMER: NEG R3 ;SET INDEX TO OTHER STATE
2646 010650 005501 ADD #1,R0 ;INCREMENT TICK COUNT
2647 010652 022701 000003 ADC R1
2648 010656 001410 CMP #3,R1 ;BRANCH IF TIMER OVERFLOWS
2649 010660 000112 BEQ TIMOVF
2650 010666 010666 JMP (R2) ;RETURN
2651 010666 032777 000100 167640 .-TIMER +24
2652 010674 001362 TIMER0: BIT #100,@MR ;BRANCH IF OSC SET
2653 010676 000112 BNE TIMER
2654 010700 013704 000616 JMP (R2) ;RETURN
2655
2656 010704 004737 017760 TIMOVF: MOV EMADDR,R4 ;TYPE TEST HEADER
2657 010710 013704 000666 JSR PC,TTOUT
2658 010714 004737 017760 MOV ERADD,R4 ;GET ERROR MSG ADDRESS
2659 010720 012704 030116 JSR PC,TTOUT ;AND TYPE IT
2660 010724 004737 017760 MOV #TMS33E,R4 ;TYPE
2661 010730 000137 010526 JSR PC,TTOUT ;'TIMER OVERFLOWED'
2662 010734 000240 JMP LT31X ;GO EXIT TEST
2663
2664 ;ROUTINE TO CHECK IF TIME IS WITHIN LIMITS. IF NOT THE ROUTINE RETURNS
2665 ;WITH THE 'V' BIT SET. THE LIMITS WERE SLECTED BY DIVIDING THE TIME
2666 ;IN MICROSECONDS BY 448. THE LOWER LIMIT IS 5,500,000 USECS (5.5 SECS);
2667 ;THE UPPER LIMIT IS 9,500,000 USECS (9.5 SECS). THE 448 IS DERIVED FROM
2668 ;56 USECS/TICK TIMES THE DIVISION BY 8 BY THE TIMOK ROUTINE.
2669 010734 000240 TIMOK: NOP
2670 010736 006201 ASR R1 ;DIVIDE COUNT BY 8
2671 010740 006000 ROR R0
2672 010742 006201 ASR R1
2673 010744 006000 ROR R0
2674 010746 006201 ASR R1
2675 010750 006000 ROR R0

```



```

2676 010752 013701 001010      MOV      SLVTYP,R1      ;..B GET SLAVE TYPE (0/1 = TE16/TU77)
2677 010756 006301              ASL      R1              ;..B FORM INDEX
2678 010760 020061 011064      CMP      R0,200$(R1)    ;..B BRANCH IF GREATER THAN LOWER LIMIT(5.5 SECS)
2679 010764 101016              BHI     1$
2680 010766 013704 000616      MOV      EMADDR,R4     ;GET ERROR MSG HEADER
2681 010772 004737 017760      JSR     PC,TTOUT       ;TYPE ERROR MSG HEADER
2682 010776 013704 000666      MOV      ERADD,R4     ;GET ERROR DESCRIPTOR MSG
2683 011002 004737 017760      JSR     PC,TTOUT
2684 011006 012704 030051      MOV      @TMS33C,R4   ;TYPE 'OCCURED TOO SOON'
2685 011012 004737 017760      JSR     PC,TTOUT
2686 011016 000262              SEV
2687 011020 000420              BR      2$
2688
2689 011022 020061 011070      1$:     CMP      R0,201$(R1) ;..B BRANCH IF LESS THAN UPPER LIMIT(9.5 SECS)
2690 011026 003415              BLE     2$
2691 011030 013704 000616      MOV      EMADDR,R4     ;GET ERROR MSG HEADER
2692 011034 004737 017760      JSR     PC,TTOUT
2693 011040 013704 000666      MOV      ERADD,R4
2694 011044 004737 017760      JSR     PC,TTOUT       ;TYPE ERROR MSG HEADER
2695 011050 012704 030073      MOV      @TMS33D,R4   ;TYPE 'OCCURED TOO LATE'
2696 011054 004737 017760      JSR     PC,TTOUT
2697 011060 000262              SEV
2698 011062 000207      2$:     RTS      PC
2699
2700
2701      ;..B TABLE OF MIN AND MAX TIMES FOR OPI FOR TE16 AND TU77 SLAVES
2702      ;..B MIN TIMES (5.5 SECS)
2703 011064 027764      200$:   .WORD   12276.    ;..B TE16
2704 011066 020622      .WORD   8594.         ;..B TU77
2705
2706      ;..B MAX TIMES (9.5 SECS)
2707 011070 051325      201$:   .WORD   21205.    ;..B TE16
2708 011072 034774      .WORD   14844.        ;..B TU77

```

2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742

011074 012737 026004 000616
011102 012737 011110 000706
011110 004737 016754
011114 013700 000660
011120 012701 040000
011124 032777 000004 167404
011132 001402
011134 052701 004000
011140 005100
011142 042700 177770
011146 052700 000300
011152 010077 167364
011156 032777 002000 167352
011164 001030
011166 012777 000071 167314
011174 017702 167324
011200 030102
011202 001011
011204 012737 030136 000666
011212 012737 000001 000712
011220 004737 015222
011224 000404
011226 020102
011230 001402
011232 004737 015210
011236 004737 016600
011242 004737 015670
011246 000137 002530

```

;LOGIC TEST 32: UNSAFE(UNS)*****
LT32:  MOV    #MSLT32,EMADDR ;SET TEST HEADER
      MOV    #LT32IT,SCOLP  ;SET SCOPE ADDRESS
LT32IT: JSR    PC,INIT3     ;INIT, SELECT DRIVE +SLAVE
      MOV    SLVN,R0        ;GET SLAVE NUMBER
      MOV    #40000,R1      ;..B SET TEST WORD (UNS)
      BIT    #4,@DT        ;..B BRANCH IF TE16
      BEQ    1$            ;..B
      BIS    #4000,R1      ;..B SET ALSO NEF FOR TU77
      COM    R0            ;SET NONEXISTANT SLAVE
      BIC    #177770,R0    ;MASK SLAVE NUMBER
      BIS    #300,R0       ;SET FORMAT
      MOV    R0,@TC        ;SELECT ILLEGAL SLAVE
      BIT    #2000,@DT     ;EXIT TEST IF SALVE AVAILABLE
      BNE    LT32XX
      MOV    #71,@C1       ;LOAD READ+GO
      MOV    @ER,R2        ;READ ER
      BIT    R1,R2         ;SEE IF EXPT=RCVD
      BNE    2$            ;IF SO: BR
      MOV    #TMS34,ERADD  ;SET ERROR CODE
      MOV    #1,EXFL       ;SET ERROR CODE
      JSR    PC,LTGERO     ;GO PRINT ERROR
      BR     LT32X
      CMP    R1,R2         ;SEE IF UNEXPECTED ERRORS
      BEQ    LT32X        ;IF NOT: BR
      JSR    PC,LTGER3    ;ELSE PRINT ERROR
      JSR    PC,ITER      ;GO SEE IF ITERATIONS
      JSR    PC,DRVCLR    ;
      JMP    TSCD2        ;RETURN TO SCHED
  
```


2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764

011252 012737 026040 000616
011260 012737 011266 000706
011266 004737 016754
011272 012777 000013 167234
011300 012777 177777 167210
011306 012777 000031 167174
011314 032777 020000 167200
011322 001010
011324 012737 027631 000666
011332 012737 000001 000712
011340 004737 015222
011344 004737 016600
011350 000137 002530

LT33: MOV #MSLT33,EMADDR
MOV #LT33IT,SCOLP
LT33IT: JSR PC,INIT3
MOV #13,@MR
MOV #-1,@FC
MOV #31,@C1
BIT #20000,@DS
BNE LT33X
MOV #TMS14,ERADD
MOV #1,EXFL
LT33X: JSR PC,LTGERO
JSR PC,ITER
JMP TSCD2

;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)*****

;SET TEST HEADER
;SET SCOPE ADDRESS
;INIT, SELECT DRIVE+SLAVE
;SET WAM 2
;SET FCS
;LOAD SPACE FORWARD.GO
;SEE IF PIP=1
;IF SO: BR
;SET ERROR CODE
;SET ERROR CODE
;GO PRINT ERROR
;GO SEE IF ITERATIONS
;RETURN TO SCHED

```

2766
2767
2768
2769 011354 012737 027551 000666 LT34: MOV #TMS6,ERADD ;SET ERROR CODE
2770 011362 012737 026074 000616 MOV #MSLT34,EMADDR ;SET TEST HEADER
2771 011370 012700 000004 LT34IT: MOV #4,R0
2772 011374 004737 016754 LT34A1: JSR PC,INIT3 ;GO INIT, SELECT DRIVE-SLAVE
2773 011400 042777 003400 167134 BIC #3400,@TC ;SELECT NRZI
2774 011406 052777 001400 167126 BIS #1400,@TC
2775 011414 032777 000040 167100 LT34A: BIT #40,@DS ;SEE IF PES=0
2776 011422 001410 BEQ LT34B ;IF SO: BR
2777 011424 012737 000002 000712 MOV #2,EXFL ;SET RCVD-NOT EXPT
2778 011432 012737 011374 000706 MOV #LT34A1,SCOLP ;SET SCOPE ADDRESS
2779 011440 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2780 011444 004737 016770 LT34B: JSR PC,INIT4
2781 011450 032777 000040 167044 LT34C: BIT #40,@DS ;SEE IF PES=1
2782 011456 001010 BNE LT34X ;IF SO: BR
2783 011460 012737 011450 000706 MOV #LT34C,SCOLP ;SET SCOPE ADDRESS
2784 011466 012737 000001 000712 MOV #1,EXFL ;SET EXPT-NOT RCVD FLAG
2785 011474 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2786 011500 004737 016600 LT34X: JSR PC,ITER ;GO SEE IF ITERATION
2787 011504 000137 002530 LT34XX: JMP TSCD2 ;RETURN TO SCHED

```



```

2789
2790
2791
2792 011510 012737 030161 000666 LT35: MOV @TMS37,ERADD
2793 011516 012737 026130 000616 MOV @MSLT35,EMADDR
2794 011524 004737 016754 LT35IT: JSR PC,INIT3 ;INIT SELECT DRIVE, SLAVE
2795 011530 032777 000020 166764 1$: BIT @20,@DS ;SEE IF SDWN IS RESET
2796 011536 001374 BNE 1$ ;IF NOT: BR
2797 011540 052777 000300 166774 BIS @300,@TC ;SET FORMAT
2798 011546 012777 000015 166760 MOV @15,@MR ;SET WAM 3
2799 011554 012777 000071 166726 MOV @71,@C1 ;LOAD READ.GO
2800 011562 032777 020000 166752 BIT @20000,@TC ;SEE IF SAC=0
2801 011570 001410 BEQ LT35A ;IF SO: BR
2802 011572 012737 000002 000712 MOV @2,EXFL ;SET RCV-NOT EXPT FLAG
2803 011600 012737 011524 000706 MOV @LT35IT,SCOLP ;SET SCOPE ADDRESS
2804 011606 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2805 011612 004737 016754 LT35A: JSR PC,INIT3 ;INIT
2806 011616 005277 166720 INC @TC ;BUMP SLAVE ADDRESS
2807 011622 032777 020000 166712 BIT @20000,@TC ;SEE IF SAC=1
2808 011630 001010 BNE LT35X ;IF SO: BR
2809 011632 012737 011612 000706 MOV @LT35A,SCOLP ;SET SCOPE ADDRESS
2810 011640 012737 000001 000712 MOV @1,EXFL ;SE EXPT-NOT RCVD FLAG
2811 011646 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2812 011652 004737 016600 LT35X: JSR PC,ITER
2813 011656 000137 002530 JMP TSCD2 ;RETURN TO SCHED

```

```

2815
2816
2817
2818 011662 012737 026175 000616 LT36: MOV #MSLT36,EMADDR
2819 011670 012737 030167 000666 MOV #TMS38,ERADD ;SET ERROR CODE
2820 011676 004737 016754 000666 LT36IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2821 011702 032777 040000 166632 BIT #40000,@TC ;SEE IF FCS=0
2822 011710 001410 BEQ 1$ ;IF SO: BR
2823 011712 012737 011676 000706 MOV #LT36IT,SCOLP ;SET SCOPE ADDRESS
2824 011720 012737 000002 000712 MOV #2,EXFL ;SET RCVD-NOT EXPT
2825 011726 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2826 011732 004737 016754 1$: JSR PC,INIT3 ;INIT
2827 011736 005077 166554 CLR @FC ;WRITE TO FC
2828 011742 032777 040000 166572 BIT #40000,@TC ;SEE IF FCS=1
2829 011750 001010 BNE LT36X ;IF SO: BR
2830 011752 012737 011732 000706 MOV #1$,SCOLP ;SET SCOPE ADDRESS
2831 011760 012737 000001 000712 MOV #1,EXFL ;SET EXPT-NOT RCVD
2832 011766 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2833 011772 004737 016600 LT36X: JSR PC,ITER
2834 011776 000137 002530 JMP TSCD2 ;RETURN TO SCHED

```



```

2836
2837
2838
2839 012002 012737 026242 000616 LT37: MOV #MSLT37,EMADDR
2840 012010 012737 030175 000666 MOV #TMS39,ERADD ;SET ERROR CODE
2841 012016 004737 016754 LT37IT: JSR PC,INIT3 ;INIT, SELECT DRIVE*SLAVE
2842 012022 052777 000300 166512 BIS #300,@TC ;SET FORMAT
2843 012030 005777 166506 TST @TC ;SEE IF ACCL=1
2844 012034 100410 BMI LT37A ;IF SO: BR
2845 012036 012737 000001 000712 MOV #1,EXFL
2846 012044 012737 012016 000706 MOV #LT37IT,SCOLP ;SET SCOPE ADDRESS
2847 012052 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2848 012056 004737 016754 LT37A: JSR PC,INIT3 ;INIT
2849 012062 052777 000300 166452 BIS #300,@TC ;SET FORMAT
2850 012070 012777 000015 166436 MOV #15,@MR ;SET WAM 3
2851 012076 012777 000071 166404 MOV #71,@C1 ;LOAD READ*GO
2852 012104 012700 100000 MOV #100000,R0 ;SET ACCL DELAY
2853 012110 005777 166426 LT37B: TST @TC ;SEE IF ACCL=0
2854 012114 100012 BPL LT37X ;IF SO: BR
2855 012116 005300 DEC R0
2856 012120 001373 BNE LT37B ;DELAY
2857 012122 012737 012056 000706 MOV #LT37A,SCOLP ;SET SCOPE ADDRESS
2858 012130 012737 000002 000712 MOV #2,EXFL
2859 012136 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2860 012142 004737 016600 LT37X: JSR PC,ITER
2861 012146 000137 002530 JMP TSCD2 ;RETURN TO SCHED

```

```

2863
2864
2865
2866 012152 012737 012166 000706 LT40: MOV #LT40IT,SCOLP ;SET SCOPE ADDRESS
2867 012160 012737 026310 000616 MOV #MSLT40,EMADDR
2868 012166 004737 016770 LT40IT: JSR PC,INIT4 ;INIT, SELECT DRIVE+SLAVE
2869 012172 005000 CLR RO
2870 012174 005300 1$: DEC RO
2871 012176 001376 BNE 1$ ;DELAY FOR OPI RESET
2872 012200 052777 002300 166334 BIS #2300,@TC
2873 012206 012777 000007 166320 MOV #7,@MR ;SET WAM 0
2874 012214 012777 000027 166266 MOV #27,@C1 ;LOAD WRITE TAPE MARK+GO
2875 012222 012700 100000 MOV #100000,RO ;SET DELAY
2876 012226 032777 000004 166266 2$: BIT #4,@DS ;SEE IF TM=1
2877 012234 001012 BNE LT40X ;IF SO: BR
2878 012236 005300 DEC RO
2879 012240 001372 BNE 2$ ;DELAY
2880 012242 012737 027527 000666 MOV #TMS3,ERADD
2881 012250 012737 000001 000712 MOV #1,EXFL
2882 012256 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2883 012262 004737 016600 LT40X: JSR PC,ITER
2884 012266 000137 002530 LT40XX: JMP TSCD2 ;RETURN TO SCHED

```



```

2886
2887
2888
2889 012272 012737 012306 000706 LT41: MOV #LT41IT,SCOLP ;SET SCOPE ADDRESS
2890 012300 012737 026355 000616 MOV #MSLT41,EMADDR
2891 012306 004737 016754 LT41IT: JSR PC,INIT3 ;INIT, SELECT DRIVE,SLAVE
2892 012312 052777 001700 166222 BIS #1700,@TC ;SET NRZ-NORMAL FORMAT
2893 012320 012777 177760 166170 MOV #-20,@FC ;SET FCS
2894 012326 012777 000007 166200 MOV #7,@MR ;SET WAM 0
2895 012334 012777 000027 166146 MOV #27,@C1 ;LOAD WRITE TAPE MARK+GO
2896 012342 005000 CLR RO
2897 012344 032777 000004 166150 1$: BIT #4,@DS ;SEE IF TM=1
2898 012352 001012 BNE 2$ ;IF SO: BR
2899 012354 005300 DEC RO
2900 012356 001372 BNE 1$ ;DELAY
2901 012360 012737 027527 000666 MOV #TMS3,ERADD ;SET ERROR CODE
2902 012366 012737 000001 000712 MOV #1,EXFL
2903 012374 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2904 012400 032777 002000 166116 2$: BIT #2000,@ER ;SEE IF ITM=1
2905 012406 001010 BNE 3$ ;IF SO: BR
2906 012410 012737 027772 000666 MOV #TMS30,ERADD ;SET ERROR CODE
2907 012416 012737 000001 000712 MOV #1,EXFL
2908 012424 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2909 012430 032777 000100 166066 3$: BIT #100,@ER ;SEE IF VPE=1
2910 012436 001011 BNE 4$ ;IF SO: BR
2911 012440 012737 027757 000666 MOV #TMS28,ERADD ;SET ERROR CODE
2912 012446 012737 000001 000712 MOV #1,EXFL
2913 012454 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
2914 012460 000410 BR LT41X
2915 012462 012701 002100 4$: MOV #2100,R1 ;SET EXPT ERROR BITS
2916 012466 017702 166032 MOV @ER,R2 ;GET ERROR REGISTER
2917 012472 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2918 012474 001402 BEQ LT41X ;IF NOT: BR
2919 012476 004737 015210 JSR PC,LTGER3 ;ELSE PRINT ERROR
2920 012502 005002 CLR R2 ;SET TIMER
2921 012504 032777 000200 166010 1$: BIT #200,@DS ;SEE IF DRY SET
2922 012512 001002 BNE 2$ ;IF SO: BR
2923 012514 005302 DEC R2 ;AWAIT DRY
2924 012516 001372 BNE 1$ ;DELAY
2925 012520 004737 016600 2$: JSR PC,ITER ;GO SEE IF ITERATIONS
2926 012524 004737 015670 JSR PC,DRVCLR ;GO DO DRIVE CLEAR
2927 012530 000137 002530 JMP TSCD2 ;RETURN TO SCHED

```

```

2929
2930 ;THE FOLLOWING SIX(6) TEST WILL REQUIRE TAPE MOVEMENT. EACH
2931 ;TEST WILL PERFORM A TAPE WRITE WHILE IN A PARTICULAR MAINTENANCE
2932 ;MODE IN ORDER TO FORCE THE REMAINING ERROR CONDITIONS.
2933
2934 ;LOGIC TEST 42: CYCLIC REDUNDANCY ERROR(CRC)*****
2935
2936 012534 012737 001700 000772 LT42: MOV #1700,UDES ;SET UNIT DESCRIPTION = NRZ
2937 012542 004737 015024 JSR PC,STATIC ;GO SEE IF STATIC ONLY
2938 012546 012700 001000 MOV #1000,R0
2939 012552 005300 1$: DEC R0
2940 012554 001376 BNE 1$ ;PAUSE
2941 012556 012737 026424 000616 MOV #MSLT42,EMADDR
2942 012564 012737 012572 000706 MOV #LT42IT,SCOLP ;SET SCOPE ADDRESS
2943 012572 004737 017004 LT42IT: JSR PC,INIT ;INIT SELECT DRIVE-SLAVE
2944 012576 012777 177770 165706 MOV #-10,@WC
2945 012604 012777 177760 165704 MOV #-20,@FC ;SET FC=20
2946 012612 012777 030204 165674 MOV #WDATA,@BA ;SET BUS ADDRESS
2947 012620 012777 000007 165706 MOV #7,@MR ;SET MM CODE
2948 012626 012777 000061 165654 MOV #61,@C1 ;LOAD WRITE-GO
2949 012634 005000 CLR R0
2950 012636 032777 000200 165656 LT42A: BIT #200,@DS ;SEE IF DRY=1
2951 012644 001002 BNE LT42B ;IF SO: BR
2952 012646 005300 DEC R0
2953 012650 001372 BNE LT42A ;DELAY
2954 012652 022777 000200 165644 LT42B: CMP #200,@ER ;SEE IF LRC ERROR ONLY
2955 012660 001007 BNE LT42B1 ;IF NOT: BR
2956 012662 017702 165642 MOV @CC,R2 ;GET CHECK CHAR
2957 012666 042702 177000 BIC #177000,R2 ;MASK CRC
2958 012672 022702 000777 CMP #777,R2 ;SEE IF SETUP CRC IS CORRECT
2959 012676 001410 BEQ LT42B2 ;IF SO: BR
2960 012700 004737 015210 LT42B1: JSR PC,LTGER3 ;ELSE PRINT ERROR SETUP
2961 012704 012704 023216 MOV #MSG55,R4
2962 012710 004737 017760 JSR PC,ITOUT ;PRINT SETUP ERROR MSG
2963 012714 000137 002530 JMP TSCD2 ;RETURN TO SCHED
2964 012720 004737 017004 LT42B2: JSR PC,INIT ;GO INIT
2965 012724 012777 177770 165560 MOV #-10,@WC ;SET WC
2966 012732 012777 177760 165556 MOV #-20,@FC ;SET FC
2967 012740 012777 030204 165546 MOV #WDATA,@BA ;SET BA
2968 012746 012777 000021 165560 MOV #21,@MR ;SET MM
2969 012754 012777 000061 165526 MOV #61,@C1 ;LOAD WRITE-GO
2970 012762 005000 CLR R0
2971 012764 032777 000200 165530 LT42C: BIT #200,@DS ;SEE IF DRY
2972 012772 001002 BNE LT42D ;IF SO: BR
2973 012774 005300 DEC R0
2974 012776 001372 BNE LT42C ;AWAIT DRY
2975 013000 005777 165520 LT42D: TST @ER ;SEE IF CRC=1
2976 013004 100411 BMI LT42E ;IF SO: BR
2977 013006 012737 030153 000666 MOV #IMS36,ERADD ;SET ERROR CODE
2978 013014 012737 000001 000712 MOV #1,EXFL
2979 013022 004737 015222 JSR PC,LTGER0 ;GO PRINT ERROR
2980 013026 000410 BR LT42X
2981 013030 012701 100200 LT42E: MOV #100200,R1 ;SET EXPT ERROR BITS
2982 013034 017702 165464 MOV @ER,R2 ;GET ERROR REGISTER
2983 013040 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2984 013042 001402 BEQ LT42X ;IF NOT: BR

```



```

2985 013044 004737 015210          JSR    PC,LTGER3      ;ELSE PRINT ERROR
2986 013050 004737 016600          LT42X: JSR    PC,ITER   ;DO ITERATIONS
2987 013054 004737 015670          JSR    PC,DRVCLR
2988 013060 000137 002530          JMP    TSCD2         ;RETURN TO SCHED
2989
2990                                     ;LOGIC TEST 43: LONGITUALINAL REDUNDANCY(LRC)*****
2991
2992 013064 032777 000004 165444 LT43:  BIT    04,0DT      ;...B BRANCH IF NOT A TE16
2993 013070 001114          BNE    LT43XX       ;...B
2994 013074 012737 001700 000772      MOV    01700,UDES   ;SET UNIT DESCRIPTION = NRZ
2995 013078 004737 015024          JSR    PC,STATIC    ;GO SEE IF STATIC ONLY
2996 013106 012737 013122 000706      MOV    0LT43IT,SCOLP ;SET SCOPE ADDRESS
2997 013114 012737 026460 000616      MOV    0MSLT43,EMADDR
2998 013122 004737 017004          LT43IT: JSR   PC,INIT  ;INIT, SELECT DRIVE-SLAVE
2999 013126 005001          CLR    R1
3000 013130 005301          18:    DEC   R1      ;DELAY
3001 013132 001376          BNE    18
3002 013134 012777 000023 165372      MOV    023,0MR      ;SET MM
3003 013142 012777 177770 165342      MOV    0-10,0WC     ;SET WC
3004 013150 012777 177760 165340      MOV    0-20,0FC     ;SET FC
3005 013156 012777 030204 165330      MOV    0WDATA,0BA   ;SET BA
3006 013164 012777 000061 165316      MOV    061,0C1      ;LOAD WRITE-GO
3007 013172 005000          CLR    R0
3008 013174 032777 000200 165320 LT43C: BIT    0200,0DS   ;SEE IF DRY
3009 013202 001002          BNE    LT43D       ;IF SO: BR
3010 013204 005300          DEC   R0
3011 013206 001372          BNE    LT43C       ;AWAIT DRY
3012 013210 032777 000200 165306 LT43D: BIT    0200,0ER   ;SEE IF LRC-1
3013 013216 001011          BNE    LT43E       ;IF SO: BR
3014 013220 012737 027743 000666      MOV    0TMS26,ERADD ;SET ERROR CODE
3015 013226 012737 000001 000712      MOV    01,EXFL
3016 013234 004737 015222          JSR    PC,LTGERO    ;GO PRINT
3017 013240 000425          BR    LT43X
3018 013242 017702 165266          LT43E: MOV    0MR,R2
3019 013246 042702 000177          BIC   0177,R2      ;MASK LRC
3020 013252 012701 157600          MOV    0157600,R1  ;SET EXPT LRC
3021 013256 020102          CMP   R1,R2        ;SEE IF EXPT = RCVD
3022 013260 001405          BEQ   LT43F        ;IF SO: BR
3023 013262 012737 023173 000666      MOV    0MSG53,ERADD ;SET ERROR CODE
3024 013270 004737 016336          JSR    PC,LTGER1    ;PRINT ERROR
3025 013274 017702 165224          LT43F: MOV    0ER,R2  ;GET ERROR REGISTER
3026 013300 012701 000200          MOV    0200,R1     ;SET EXPT ERROR BITS
3027 013304 020102          CMP   R1,R2        ;SEE IF UNEXPECTED ERRORS
3028 013306 001402          BEQ   LT43X        ;IF NOT: BR
3029 013310 004737 015210          JSR    PC,LTGER3    ;ELSE PRINT ERROR
3030 013314 004737 016600          LT43X: JSR    PC,ITER
3031 013320 004737 015670          JSR    PC,DRVCLR
3032 013324 000137 002530          LT43XX: JMP   TSCD2 ;RETURN TO SCHED

```

```

;LOGIC TEST 44: PE CORRECTABLE DATA (CORR)*****
3034
3035
3036 013330 012737 002300 000772 LT44: MOV @2300,UDES ;SET UNIT DESCRIPTION = PE
3037 013336 004737 015024 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3038 013342 012737 026514 000616 MOV @MSLT44,EMADDR ;SET HEADER
3039 013350 012737 013356 000706 MOV @LT44IT,SCOLP ;SET SCOP
3040 013356 004737 017004 LT44IT: JSR PC,INIT ;GO INITIALIZE
3041 013362 012777 177600 165122 MOV @-200,@WC ;SET WC=200
3042 013370 012777 177400 165120 MOV @-400,@FC ;SET FC=400
3043 013376 012777 030204 165110 MOV @WDATA,@BA ;SET BA=START OF WRITE BUFFER
3044 013404 012777 000061 165076 MOV @61,@C1 ;LOAD WRITE AND GO
3045 013412 005000 CLR RO
3046 013414 005777 165076 LT44A: TST @FC ;SEE IF FC=0
3047 013420 001402 BEQ LT44A1 ;IF SO:BR
3048 013422 005300 DEC RO
3049 013424 001373 BNE LT44A ;AWAIT FC=0
3050 013426 012777 000021 165100 LT44A1: MOV @21,@MR ;SET MAINT MODE
3051 013434 005000 CLR RO
3052 013436 032777 000200 165056 LT44B: BIT @200,@DS ;SEE IF DRY
3053 013444 001002 BNE LT44C ;IF SO :BR
3054 013446 005300 DEC RO
3055 013450 001372 BNE LT44B ;AWAIT DRY
3056 013452 005777 165046 LT44C: TST @ER ;SEE IF CORR=1
3057 013456 100410 BMI LT44D ;IF SO: BR
3058 013460 012737 030144 000666 MOV @TMS35,ERADD ;ELSE SET ERROR CODE
3059 013466 012737 000001 000712 MOV @1,EXFL ;SET EXPT FLAG
3060 013474 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
3061 013500 000240 LT44D: NOP
3062 013502 122777 000002 165020 LT44E: CMPB @2,@CC ;SEE IF DEAD TRACK BIT 1
3063 013510 001414 BEQ LT44F ;IF SO: BR
3064 013512 117702 165012 MOVB @CC,R2 ;ELSE SAVE RECVD
3065 013516 042702 177000 BIC @177000,R2 ;MASK OUT CRC
3066 013522 112701 000002 MOVB @2,R1 ;SAVE EXPT
3067 013526 012737 022602 000666 MOV @MSG42,ERADD ;SET ERROR CODE
3068 013534 004737 016336 JSR PC,LTGER1 ;GO PRINT ERROR
3069 013540 000410 BR LT44X
3070 013542 017702 164756 LT44F: MOV @ER,R2 ;GET ERROR REGISTER
3071 013546 012701 100000 MOV @100000,R1 ;SET EXPT ERROR BITS
3072 013552 020102 CMP R1,R2 ;SEE IF EXPT=RCVD
3073 013554 001402 BEQ LT44X ;IF SO: BR
3074 013556 004737 015210 JSR PC,LTGER3 ;ELSE PRINT ERROR
3075 013562 004737 016600 LT44X: JSR PC,ITER ;GO SEE IF ITERATIONS
3076 013566 004737 015670 JSR PC,DRVCLR ;GO DO DRIVE CLEAR
3077 013572 000137 002530 LT44XX: JMP TSCD2 ;RETURN TO SCHED

```



```

3079
3080          ;LOGIC TEST 45: PE INCORRECTABLE DATA(INC)*****
3081
3082 013576 012737 002300 000772 LT45:  MOV    #2300, JDES      ;SET UNIT DESCRIPTION = PE
3083 013604 004737 015024          JSR    PC, STATIC    ;GO SEE IF STATIC ONLY
3084 013610 012737 026574 000616          MOV    #MSLT45, EMADDR
3085 013616 012737 013624 000706          MOV    #LT45IT, SCOLP
3086 013624 004737 017004          LT45IT: JSR    PC, INIT      ;INIT SELECT DRIVE SLAVE
3087 013630 012777 177600 164654          MOV    #-200, @WC     ;SET WC=200
3088 013636 012777 177400 164652          MOV    #-400, @FC    ;SET FC=400
3089 013644 012777 030204 164642          MOV    #WDATA, @BA   ;SET BA=START OF WRITE BUFFER
3090 013652 012777 000061 164630          MOV    #61, @C1     ;LOAD WRITE+GO
3091 013660 005000          CLR    R0
3092 013662 005777 164630          LT45E: TST    @FC     ;AWAIT FC=0
3093 013666 001402          BEQ    LT45E1
3094 013670 005300          DEC    R0
3095 013672 001373          BNE    LT45E        ;AWAIT FC=0
3096 013674 012777 000023 164632 LT45E1: MOV    #23, @MR      ;SET MAINT CODE
3097 013702 005000          CLR    R0
3098 013704 032777 000200 164610 LT45A:  BIT    #200, @DS   ;SEE IF DRY IS SET
3099 013712 001002          BNE    LT45B        ;IF SO: BR
3100 013714 005300          DEC    R0
3101 013716 001372          BNE    LT45A        ;AWAIT DRY
3102 013720 032777 000100 164576 LT45B:  BIT    #100, @ER   ;SEE IF INC=1
3103 013726 001010          BNE    LT45D        ;IF SO:BR
3104 013730 012737 027721 000666          MOV    #TMS23, ERADD ;SET ERROR CODE
3105 013736 012737 000001 000712          MOV    #1, EXFL
3106 013744 004737 015222          JSR    PC, LTGERO   ;GO PRINT ERROR
3107 013750 017702 164554          LT45D:  MOV    @CC, R2   ;GET CHECK CHAR
3108 013754 042702 177000          BIC    #177000, R2  ;MASK CHECK CHAR
3109 013760 012701 000046          MOV    #46, R1     ;SET EXPT CK
3110 013764 020102          CMP    R1, R2      ;SEE IF EXPT = RCVD
3111 013766 001405          BEQ    LT45F        ;IF SO: BR
3112 013770 012737 023205 000666          MOV    #MSG54, ERADD
3113 013776 004737 016336          JSR    PC, LTGER1   ;ELSE GO PRINT ERROR
3114 014002 017702 164516          LT45F:  MOV    @ER, R2
3115 014006 042702 120600          BIC    #120600, R2  ;MASK OPI ,NSG, CORR, AND PEF
3116 014012 012701 000100          MOV    #100, R1    ;SET EXPT ERROR BITS
3117 014016 020102          CMP    R1, R2      ;SEE IF UNEXPECTED ERRORS
3118 014020 001402          BEQ    LT45X        ;IF NOT: BR
3119 014022 004737 015210          JSR    PC, LTGER3   ;ELSE PRINT ERROR
3120 014026 004737 016600          LT45X:  JSR    PC, ITER
3121 014032 004737 015670          JSR    PC, DRVCLR
3122 014036 000137 002530          LT45XX: JMP    TSCD2 ;RETURN TO SCHED

```

```

3124
3125
3126
3127 014042 012737 002300 000772 LT46: MOV #2300,JDES ;SET UNIT DESCRIPTION = PE
3128 014050 004737 015024 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3129 014054 012737 026656 000616 MOV #MSLT46,EMADDR ;SET HEADER
3130 014062 012737 014070 000706 MOV #LT46IT,SCOLP ;SET SCOPE ADDRESS
3131 014070 004737 017004 LT46IT: JSR PC,INIT ;INITIALIZE
3132 014074 012777 177770 164410 MOV #-10,@WC ;SET WC=10
3133 014102 012777 177760 164406 MOV #-20,@FC ;SET FC=20
3134 014110 012777 030204 164376 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
3135 014116 012777 000061 164364 MOV #61,@C1 ;LOAD WRITE*GO
3136 014124 005777 164366 LT46A: TST @FC
3137 014130 001375 BNE LT46A ;AWAIT FC=0
3138 014132 032777 000100 164374 1$: BIT #100,@MR ;WAIT FOR TAPE TO START WRITEING POSTAMBLE
3139 014140 001774 BEQ 1$ ;DELAY
3140 014142 032777 000100 164364 2$: BIT #100,@MR
3141 014150 001374 BNE 2$
3142 014152 012777 000027 164354 MOV #27,@MR ;SET MM CODE TO KILL PEF
3143 014160 005000 CLR R0 ;INIT TIMING LOOP
3144 014162 032777 000200 164332 LT46B: BIT #200,@DS ;SEE IF DRY SET
3145 014170 001002 BNE LT46C ;IF SO: BR
3146 014172 005300 DEC R0
3147 014174 001372 BNE LT46B ;AWAIT DRY
3148 014176 032777 000200 164320 LT46C: BIT #200,@ER ;SEE IF PEF SET
3149 014204 001011 BNE LT46D ;IF SO: BR
3150 014206 012737 027735 000666 MOV #TMS25,ERADD ;SET ERROR TAG
3151 014214 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLAG
3152 014222 004737 015222 JSR PC,LTGERO ;GO PRINT ERROR
3153 014226 000420 BR LT46X
3154 014230 017702 164270 LT46D: MOV @ER,R2 ;GET ERROR REGISTER
3155 014234 042702 120100 BIC #120100,R2 ;**B CLEAR CRC,OPI & INC BITS (MAY OR MAY NOT SET)
3156 014240 012701 000600 MOV #600,R1 ;**B SET EXPT ERROR BITS (NSG + PEF)
3157 014244 032777 000004 164264 BIT #4,@DT ;**B BRANCH IF TE16
3158 014252 001402 BEQ 1$ ;**B
3159 014254 042701 000400 BIC #400,R1 ;**B TU77 SHOULD NOT SET NSG BIT
3160 014260 020102 1$: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3161 014262 001402 BEQ LT46X ;IF NOT: BR
3162 014264 004737 015210 JSR PC,LTGER3 ;ELSE PRINT ERROR
3163 014270 004737 016600 LT46X: JSR PC,ITER
3164 014274 004737 015670 JSR PC,DRVCLR
3165 014300 000137 002530 LT46XX: JMP TSCD2 ;RETURN TO SCHED

```



```

3167                                     ;LOGIC TEST 47: FRAME COUNT OVERFLOW(M8905-YB)*****
3168
3169 014304 012737 026712 000616 LT47:  MOV    #MSLT47,EMADDR ;SET TEST HEADER
3170 014312 012737 014320 000706      MOV    #LT47IT,SCOLP ;SET SCOPE ADDRESS
3171 014320 004737 016754              LT47IT: JSR    PC,INIT3 ;GO INIT
3172 014324 012777 177770 164160      MOV    #-10,@WC ;SET WC = 10
3173 014332 012777 177760 164156      MOV    #-20,@FC ;SET FC = 20
3174 014340 052777 001700 164174      BIS    #1700,@TC ;SET TO NRZ, NORMAL, ODD
3175 014346 012777 030204 164140      MOV    #WDATA,@BA ;SET BUS ADDRESS
3176 014354 012777 000013 164152      MOV    #13,@MR ;SET WRAP 2
3177 014362 012777 000061 164120      MOV    #61,@C1 ;LOAD WRITE+GO
3178 014370 012700 040000              MOV    #40000,R0
3179 014374 005777 164142              LT47A: TST    @TC ;SEE IF ALPHA
3180 014400 100002                      BPL    LT47B ;IF SO: BR
3181 014402 005300                      DEC    R0
3182 014404 001373                      BNE    LT47A ;AWAIT ALPHA
3183 014406 012700 000020              LT47B: MOV    #20,R0 ;SET CLK CNT
3184 014412 052777 000040 164114      LT47C: BIS    #40,@MR
3185 014420 042777 000040 164106      BIC    #40,@MR ;CLOCK MR
3186 014426 005300                      DEC    R0
3187 014430 001370                      BNE    LT47C ;IF NOT DONE ALL: BR
3188 014432 017702 164060              MOV    @FC,R2
3189 014436 005001                      CLR    R1 ;SET TEST WORD
3190 014440 020102                      CMP    R1,R2 ;SEE IF EXPT = RCVD
3191 014442 001410                      BEQ    LT47X ;IF SO: BR
3192 014444 012737 022055 000666      MOV    #MSG19,ERADD ;SET ERROR CODE
3193 014452 012737 000001 000712      MOV    #1,EXFL ;SET EXPT FLAG
3194 014460 004737 016336              JSR    PC,LTGER1 ;GO PRINT ERROR
3195 014464 004737 016600              LT47X: JSR    PC,ITER ;GO SEE IF ITERATIONS
3196 014470 000137 002530              JMP    TSCD2 ;RETURN TO SCHEDULAR
3197

```

```

3199
3200 ;LOGIC TEST 50: NEF WHEN WRITING PE ON NRZ SELECTED SLAVE
3201
3202 014474 012737 026762 000616 LT50: MOV #MSLT50,EMADDR ;SET ERROR POINTER FOR STANDARD
3203 014502 012737 014510 000706 MOV #LT50IT,SCOLP
3204 014510 004737 016754 LT50IT: JSR PC,INIT3 ;SET SLAVE = NRZ
3205 014514 042777 003400 164020 BIC #3400,@TC ;CLEAR DENSITY BITS
3206 014522 052777 002300 164012 BIS #2300,@TC ;SET DENSITY = PE
3207 014530 012777 177770 163754 MOV #-10,@WC ;SET WORD COUNT
3208 014536 012777 177760 163752 MOV #-20,@FC ;SET FRAME COUNT
3209 014544 012777 030204 163742 MOV #WDATA,@BA ;SET BUS ADDRESS
3210 014552 012777 000013 163754 MOV #13,@MR ;SET WRAP 2
3211 014560 012777 000061 163722 MOV #61,@C1 ;LOAD WRITE COMMAND
3212 014566 000240 NOP
3213 014570 000240 NOP
3214 014572 000240 NOP
3215 014574 012701 004000 2$: MOV #4000,R1 ;SET EXPECTED RESULT
3216 014600 017702 163720 3$: MOV @ER,R2 ;GET ERROR REGISTER
3217 014604 030102 BIT R1,R2 ;BRANCH IF NEF BIT SET
3218 014606 001006 BNE 1$
3219 014610 012737 000001 000712 MOV #1,EXFL ;SET EXPECTED FLAG
3220 014616 004737 015222 JSR PC,LTGER0 ;PRINT ERROR
3221 014622 000404 BR LT50X
3222 014624 020102 1$: CMP R1,R2 ;BRANCH IF NO UNEXPECTED ERROR
3223 014626 001402 BEQ LT50X ;BITS WERE SET
3224 014630 004737 015210 JSR PC,LTGER3 ;PRINT ERROR MSG
3225 014634 004737 016600 LT50X: JSR PC,ITER ;ITERATE TEST
3226 014640 004737 015670 JSR PC,DRVCLR ;RESET DRIVE
3227 014644 000137 002530 JMP TSCD2
3228
3229

```



```

3231
3232
3233
3234 014650 012737 027042 000616 LT51: MOV #MSLT51,EMADDR ;•SET ERROR POINTER FOR STANDARD CONF.
3235 014656 012737 014664 000706 MOV #LT51IT,SCOLP ;SET SCOPE LOOP ADDRS.
3236 014664 004737 016770 LT51IT: JSR PC,INIT4 ;SET SLAVE = PE
3237 014670 042777 002300 163644 BIC #2300,@TC ;CLEAR DENSITY BITS
3238 014676 052777 001300 163636 BIS #1300,@TC ;SET DENSITY = NRZ
3239 014704 012777 177770 163600 MOV #-10,@WC ;SET WORD COUNT
3240 014712 012777 177760 163576 MOV #-20,@FC ;SET FRAME COUNT
3241 014720 012777 030204 163566 MOV #WDATA,@BA ;SET BUS ADDRESS
3242 014726 012777 000013 163600 MOV #13,@MR ;SET WRAP 2
3243 014734 012777 000061 163546 MOV #61,@C1 ;SET WRITE COMMAND AND GO
3244 014742 000240 NOP
3245 014744 000240 NOP
3246 014746 000240 NOP
3247 014750 012701 004000 MOV #4000,R1 ;SET EXPECTED RESULT
3248 014754 017702 163544 MOV @ER,R2 ;GET ERROR REGISTER
3249 014760 030102 BIT R1,R2 ;BRANCH IF NEF SET
3250 014762 001006 BNE 1$
3251 014764 012737 000001 000712 MOV #1,EXFL ;SET EXPECTED FLAG
3252 014772 004737 015222 JSR PC,LTGERO ;PRINT ERROR MSG
3253 014776 000404 BR LT51X
3254 015000 020102 1$: CMP R1,R2 ;BRANCH IF NO UNEXPECTED
3255 015002 001402 BEQ LT51X ;ERROR BITS WERE SET
3256 015004 004737 015210 LT51X: JSR PC,LTGER3
3257 015010 004737 016600 JSR PC,ITER ;•ITERATE TEST
3258 015014 004737 015670 JSR PC,DRVCLR ;CLEAR DRIVE
3259 015020 000137 002530 JMP TSCD2 ;RETURN TO SCHEDULER

```

```

3261
3262
3263 015024 005737 000734
3264 015030 001006
3265 015032 005737 001006
3266 015036 001403
3267 015040 005726
3268 015042 000137 002530
3269 015046 005037 001002
3270 015052 000207
3271

```

;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

```



```
3273  
3274  
3275  
3276 015054 017700 163454  
3277 015060 042700 000036  
3278 015064 052700 000024  
3279 015070 010077 163440  
3280 015074 042777 000037 163432  
3281 015102 005000  
3282 015104 012701 000002  
3283 015110 032777 000001 163372 EORP1:  
3284 015116 001430  
3285 015120 005300  
3286 015122 001372  
3287 015124 005301  
3288 015126 001370  
3289 015130 032777 020000 163432  
3290 015136 001020  
3291 015140 005737 000614  
3292 015144 001004  
3293 015146 013704 000616  
3294 015152 004737 017760  
3295 015156 012704 030617  
3296 015162 004737 017760  
3297 015166 032777 100000 163374  
3298 015174 001401  
3299 015176 000000  
3300 015200 000240  
3301 015202 005037 000672  
3302 015206 000207  
3303
```

```
      ;END OF RECORD FORCE SUBROUTINE*****  
EORPA: MOV @MR,R0 ;GET MAINT REG  
      BIC #36,R0 ;CLEAR CURRENT OP CODE  
      BIS #24,R0 ;SET EOR CLEAR OP CODE  
      MOV R0,@MR ;DO EOR  
      BIC #37,@MR ;CLEAR EOR AND MM  
      CLR R0  
      MOV #2,R1  
EORP1: BIT #1,@C1 ;SEE IF GO GONE  
      BEQ EORP2 ;IF SO: BR  
      DEC R0 ;AWAIT GO RESET  
      BNE EORP1  
      DEC R1  
      BNE EORP1  
      BIT #20000,@SWR ;SEE IF ERROR PRINT INHIBIT  
      BNE EORP2 ;IF SO: BR  
      TST HDRFL ;SEE IF DONE HEADER  
      BNE EORP1A ;IF SO: BR  
      MOV EMADDR,R4  
      JSR PC,TTOUT ;PRINT HEADER  
EORP1A: MOV @WMSG31,R4 ;PRINT EOR GO BIT ERROR  
      JSR PC,TTOUT ;SEE IF HALT ON ERROR  
      BIT #100000,@SWR ;IF NOT: BR  
      BEQ EORP2  
      HALT  
EORP2: NOP ;CLEAR FLAG  
EORPX: CLR TEMP2 ;RETURN  
      RTS PC
```

```

3305 ;LOGIC TEST ADDRESSING ERROR SUBROUTINE*****
3306
3307 015210 005037 000712 LTGER3: CLR EXFL
3308 015214 012737 023144 000666 MOV #MSG51,ERADD
3309 015222 012737 000001 000742 LTGER0: MOV #1,ADDFL ;SET NO ADDRESS FLAG
3310 015230 000240 LTGER: NOP
3311 015232 005037 000662 CLR PFLG ;CLEAR PRINT FLAG
3312 015236 032777 020000 163324 BIT #20000,@SWR ;SEE IF SHOULD PRINT
3313 015244 001112 BNE LTGX ;IF NOT: BR
3314 015246 005737 000614 LTGA: TST HDRFL ;SEE IF PRINTED HEADER
3315 015252 001004 BNE LTGA1 ;IF SO: BR
3316 015254 013704 000616 MOV EMADDR,R4
3317 015260 004737 017760 JSR PC,TTOUT ;PRINT TEST HEADER
3318 015264 012737 000001 000614 LTGA1: MOV #1,HDRFL ;SET HEADER FLAG
3319 015272 013704 000666 MOV ERADD,R4
3320 015276 004737 017760 JSR PC,TTOUT ;PRINT CONDITION ERROR
3321 015302 005737 000742 TST ADDFL
3322 015306 001003 BNE LTGA2
3323 015310 010103 MOV R1,R3
3324 015312 004737 020172 JSR PC,OCTP ;PRINT ADDRESS
3325 015316 005737 000712 LTGA2: TST EXFL
3326 015322 001412 BEQ LTGC ;IF NO STATUS: BR
3327 015324 012704 021435 MOV #MSG6,R4
3328 015330 022737 000001 000712 CMP #1,EXFL ;EXPT-NOT RCVD
3329 015336 001402 BEQ LTGB
3330 015340 012704 021454 MOV #MSG7,R4 ;RCVD-NOT EXPT
3331 015344 004737 017760 LTGB: JSR PC,TTOUT ;PRINT STATUS
3332 015350 005237 000662 LTGC: INC PFLG
3333 015354 005737 000742 TST ADDFL ;SEE IF ADD TST
3334 015360 001430 BEQ LTGD ;IF SO: BR
3335 015362 005737 000740 TST T24FL ;SEE IF TEST 24
3336 015366 001423 BEQ LTGCO ;IF NOT: BR
3337 015370 012704 030604 MOV #WMSG27,R4
3338 015374 004737 017760 JSR PC,TTOUT ;PRINT DATA TAG
3339 015400 012704 021755 MOV #MSG12,R4
3340 015404 004737 017760 JSR PC,TTOUT ;PRINT EXPT TAG
3341 015410 012703 177777 MOV #-1,R3
3342 015414 004737 020162 JSR PC,OCTPE ;PRINT EXPT
3343 015420 012704 021764 MOV #MSG13,R4
3344 015424 004737 017760 JSR PC,TTOUT ;PRINT RCVD TAG
3345 015430 010103 MOV R1,R3 ;GET RCVD
3346 015432 004737 020162 JSR PC,OCTPE ;PRINT RCVD
3347 015436 004737 015534 LTGCO: JSR PC,REGP ;PRINT REGISTERS
3348 015442 032777 004000 163120 LTGD: BIT #4000,@SWR
3349 015450 001010 BNE LTGX
3350 015452 012704 022026 MOV #MSG16,R4
3351 015456 004737 017760 JSR PC,TTOUT
3352 015462 013703 000676 MOV ITCNT,R3 ;PRINT ITERATION
3353 015466 004737 020172 JSR PC,OCTP
3354 015472 005777 163072 LTGX: TST @SWR
3355 015476 100001 BPL LTGXA ;IF NOT STOP ON ERROR: BR
3356 015500 000000 HALT
3357 015502 005737 000662 LTGXA: TST PFLG
3358 015506 001004 BNE LTGXX ;IF PRINTED: BR
3359 015510 032777 020000 163052 BIT #20000,@SWR
3360 015516 001653 BEQ LTGA

```


3361 015520 005037 000742
3362 015524 005037 000712
3363 015530 000137 016550
3364
3365
3366
3367 015534 000240
3368 015536 012704 022767
3369 015542 004737 017760
3370 015546 017703 162736
3371 015552 004737 020162
3372 015556 017703 162730
3373 015562 004737 020162
3374 015566 017703 162722
3375 015572 004737 020162
3376 015576 017703 162714
3377 015602 004737 020162
3378 015606 017703 162706
3379 015612 004737 020162
3380 015616 017703 162700
3381 015622 004737 020162
3382 015626 017703 162672
3383 015632 004737 020162
3384 015636 017703 162664
3385 015642 004737 020162
3386 015646 017703 162662
3387 015652 004737 020162
3388 015656 017703 162660
3389 015662 004737 020162
3390 015666 000207
3391
3392

LTGXX: CLR ADDFL ;CLEAR ADDRESS FLAG
CLR EXFL
JMP SCOPE

;SUBROUTINE TO PRINT MAJOR REGISTERS*****

REGP: NOP
MOV @MSG46,R4
JSR PC,TTOUT ;PRINT REGISTER HEADER

MOV @C1,R3
JSR PC,OCTPE
MOV @WC,R3
JSR PC,OCTPE
MOV @BA,R3
JSR PC,OCTPE
MOV @FC,R3
JSR PC,OCTPE
MOV @CS,R3
JSR PC,OCTPE

;PRINT REGISTERS

MOV @DS,R3
JSR PC,OCTPE
MOV @ER,R3
JSR PC,OCTPE
MOV @AS,R3
JSR PC,OCTPE
MOV @MR,R3
JSR PC,OCTPE
MOV @TC,R3
JSR PC,OCTPE
RTS PC

```

3394                                     ;DRIVE CLEAR SUBROUTINE*****
3395
3396 015670 000240          DRVCLR: NOP
3397 015672 012704 040000      MOV      #40000,R4
3398 015676 005304          DCD:   DEC      R4
3399 015700 001376          BNE     DCD                ;DELAY
3400 015702 005037 000662      CLR     PFLG
3401 015706 004737 016124      JSR     PC,ATTN           ;GO SEE OF ATTN SET
3402 015712 012777 000011 162570  MOV     #11,@C1         ;ISSUE DRIVE CLEAR
3403 015720 005000          CLR     R0
3404 015722 032777 000200 162572  DCA:   BIT     #200,@DS        ;SEE IF DRY
3405 015730 001002          BNE     DCA0
3406 015732 005300          DEC     R0
3407 015734 001372          BNE     DCA                ;WAIT FOR DRY
3408 015736 032777 040000 162556  DCA0:  BIT     #40000,@DS      ;SEE IF ERR RESET
3409 015744 001022          BNE     DCE                ;IF NOT: BR
3410 015746 005777 162552      TST     @ER                ;SEE IF ERROR REGISTER RESET
3411 015752 001017          BNE     DCE                ;IF NOT: BR
3412 015754 005777 162542      TST     @DS                ;SEE IF ATA RESET
3413 015760 100414          BMI     DCE                ;IF NOT: BR
3414 015762 012703 000001      MOV     #1,R3              ;SET TEST BIT
3415 015766 013704 000620      MOV     DRVN,R4            ;GET DRIVE NUMBER & BRANCH
3416 015772 001403          BEQ     DCC                ;IF DRIVE 0
3417 015774 006303          DCB:   ASL     R3              ;POSITION TEST BIT PER DRIVE NUMBER
3418 015776 005304          DEC     R4                ;SEE IF DONE
3419 016000 001375          BNE     DCB                ;IF NOT: BR
3420 016002 030377 162520      DCC:   BIT     R3,@AS        ;SEE IF ATTEN IS RESET
3421 016006 001001          BNE     DCE                ;IF NOT: BR
3422 016010 000207          RTS     PC                ;RETURN
3423
3424 016012 000240          DCE:   NOP
3425 016014 032777 020000 162546  BIT     #20000,@SWR        ;SEE IF ERROR PRINT INHIBIT
3426 016022 001017          BNE     DCEX              ;IF SO: BR
3427 016024 005737 000614      TST     HDRFL              ;SEE IF PRINT HEADER
3428 016030 001004          BNE     DCEA              ;IF NOT: BR
3429 016032 013704 000616      MOV     EMADDR,R4
3430 016036 004737 017760      JSR     PC,TTOUT           ;PRINT HEADER
3431 016042 012704 023073          DCEA:  MOV     #MSG47,R4
3432 016046 004737 017760      JSR     PC,TTOUT           ;PRINT DRIVE CLEAR ERROR
3433 016052 004737 015534      JSR     PC,REGP           ;PRINT REGISTERS
3434 016056 005237 000662      INC     PFLG              ;SET PRINTED FLAG
3435 016062 005777 162502          DCEX:  TST     @SWR          ;SEE IF HALT ON ERROR
3436 016066 100001          BPL     DCEXA            ;IF NOT: BR
3437 016070 000000          HALT
3438 016072 005737 000662          DCEXA: TST     PFLG          ;SEE IF HAVE PRINTED
3439 016076 001004          BNE     DCEXX            ;IF SO: BR
3440 016100 032777 020000 162462  BIT     #20000,@SWR        ;BRANCH IF ERROR
3441 016106 001741          BEQ     DCE                ;PRINTOUT DESIRED
3442 016110 000240          DCEXX: NOP
3443 016112 012737 015670 000706  MOV     #DRVCLR,SCOLP      ;SET SCOPE LOOP ADDRESS
3444 016120 000137 016550      JMP     SCOPE              ;GO DO SCOPE LOOP

```



```

3446                                     ;COMPOSITE ERROR CHECK SUBROUTINE*****
3447
3448 016124 000240                       ATTN:  NOP
3449 016126 005777 162370                TST     @DS          ;SEE IF ATA SET
3450 016132 001004                       BNE     ATTA        ;IF SO: BR
3451 016134 012737 022431 000674        MOV     #MSG32,TEMP3
3452 016142 000427                       BR      ATTP        ;ELSE PRINT ERROR
3453 016144 032777 040000 162350        ATTA:  BIT     #40000,@DS ;SEE IF COMPOSITE ERROR SET
3454 016152 001004                       BNE     ATTB        ;IF SO: BR
3455 016154 012737 022413 000674        MOV     #MSG31,TEMP3
3456 016162 000417                       BR      ATTP        ;ELSE PRINT ERROR
3457 016164 012703 000001                ATTB:  MOV     #1,R3  ;SET TEST BIT
3458 016170 012737 022447 000674        MOV     #MSG33,TEMP3
3459 016176 013704 000620                MOV     DRVN,R4    ;GET DRIVE NUMBER & BRANCH
3460 016202 001403                       BEQ     ATTD        ;IF DRIVE 0
3461 016204 006303                       ATTC:  ASL     R3    ;POSITION TEST BIT
3462 016206 005304                       DEC     R4         ;SEE IF DONE
3463 016210 001375                       BNE     ATTC        ;IF NOT: BR
3464 016212 030377 162310                ATTD:  BIT     R3,@AS ;SEE IF ATTEN SUMMARY SET
3465 016216 001401                       BEQ     ATTP        ;IF NOT: BR
3466 016220 000207                       RTS     PC         ;ELSE RETURN
3467 016222 032777 020000 162340        ATTP:  BIT     #20000,@SWR ;SEE IF PRINT INHIBIT
3468 016230 001021                       BNE     ATTX        ;IF SO: BR
3469 016232 005737 000614                TST     HDRFL      ;SEE IF DONE HEADER
3470 016236 001004                       BNE     ATTPA       ;IF SO: BR
3471 016240 013704 000616                MOV     EMADDR,R4
3472 016244 004737 017760                JSR     PC,TTOUT   ;PRINT HEADER
3473 016250 013704 000674                ATTPA: MOV     TEMP3,R4
3474 016254 004737 017760                JSR     PC,TTOUT   ;PRINT ERROR TYPE
3475 016260 004737 015534                JSR     PC,REGP    ;PRINT REGISTERS
3476 016264 005237 000662                INC     PFLG       ;SET PRINT FLAG
3477 016270 005237 000614                INC     HDRFL      ;SET HEADER FLAG
3478 016274 005777 162270                ATTX:  TST     @SWR  ;SEE IF HALT ON ERROR
3479 016300 100001                       BPL     ATTXA       ;IF NOT: BR
3480 016302 000000                       HALT
3481 016304 005737 000662                ATTXA: TST     PFLG  ;SEE IF DONE PRINT
3482 016310 001004                       BNE     ATTXX       ;IF SO: BR
3483 016312 032777 020000 162250        BIT     #20000,@SWR ;BRANCH IF NO ERROR
3484 016320 001740                       BEQ     ATTP        ;PRINTOUT DESIRED
3485 016322 005037 000662                ATTXX: CLR     PFLG  ;CLEAR PRINT FLAG
3486 016326 000207                       RTS     PC         ;RETURN

```

```

3488                                     ;LOGIC TEST REGISTER BIT ERROR SUBROUTINE*****
3489
3490 016330 012737 000001 000732 LTGER2: MOV    #1,PEXFL      ;SET FLAG
3491 016336 000240                LTGER1: NOP
3492 016340 005037 000662                CLR    PFLG          ;CLEAR PRINT FLAG
3493 016344 032777 020000 162216        BIT    #20000,@SWR   ;BRANCH IF ERROR
3494 016352 001055                BNE    LTG1X         ;PRINTOUT DESIRED
3495 016354 005737 000614                LTG1A: TST   HDRFL   ;SEE IF PRINT HEADER
3496 016360 001004                BNE    LTG1B         ;IF NOT: BR
3497 016362 013704 000616                MOV    EMADDR,R4
3498 016366 004737 017760                JSR    PC,TTOUT     ;PRINT HEADER
3499 016372 012737 000001 000614 LTG1B: MOV    #1,HDRFL   ;SET FLAG
3500 016400 013704 000666                MOV    ERADD,R4
3501 016404 004737 017760                JSR    PC,TTOUT     ;PRINT ERROR CODE
3502 016410 005737 000732                TST   PEXFL         ;SEE IF PRINT EXPT RCVD
3503 016414 001016                BNE    LTG1T         ;IF NOT: BR
3504 016416 012704 021755                MOV    #MSG12,R4
3505 016422 004737 017760                JSR    PC,TTOUT     ;PRINT EXPT TAG
3506 016426 010103                MOV    R1,R3
3507 016430 004737 020172                JSR    PC,OCIP      ;PRINT EXPT
3508 016434 012704 021764                MOV    #MSG13,R4
3509 016440 004737 017760                JSR    PC,TTOUT     ;PRINT RCVD TAG
3510 016444 010203                MOV    R2,R3
3511 016446 004737 020172                JSR    PC,OCIP      ;PRINT RCVD
3512 016452 032777 004000 162110 LTG1T: BIT    #4000,@SWR
3513 016460 001010                BNE    LTG1C
3514 016462 012704 022026                MOV    #MSG16,R4
3515 016466 004737 017760                JSR    PC,TTOUT
3516 016472 013703 000676                MOV    ITCNT,R5
3517 016476 004737 020172                JSR    PC,OCIP      ;PRINT ITERATION
3518 016502 005237 000662                LTG1C: INC    PFLG
3519 016506 000240                LTG1X: NOP
3520 016510 005777 162054                TST   @SWR
3521 016514 100001                BPL    LTG1X1       ;IF NOT STOP ON ERROR: BR
3522 016516 000000                HALT
3523 016520 005737 000662                LTG1X1: TST   PFLG
3524 016524 001004                BNE    LTG1XX       ;IF HAVE PRINTED: BR
3525 016526 032777 020000 162034        BIT    #20000,@SWR
3526 016534 001707                BEQ    LTG1A
3527 016536 000240                LTG1XX: NOP
3528 016540 005037 000732                CLR    PEXFL        ;CLEAR EXPT-RCVD FLAG
3529 016544 000137 016550                JMP    SCOPE        ;GO TO SCOPE
3530
3531                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
3532
3533
3534 016550 000240                SCOPE: NOP
3535 016552 032777 040000 162010        BIT    #40000,@SWR  ;SEE IF LOOP ON ERROR
3536 016560 001001                BNE    1$           ;IF SO: BR
3537 016562 000207                RTS    PC           ;ELSE EXIT
3538 016564 000240                1$:  NOP
3539 016566 005726                TST   (SP)         ;RESET STACK
3540 016570 000240                NOP
3541 016572 000240                NOP
3542 016574 000177 162106                JMP    @SCOLP      ;LOOP ON ERROR
3543

```



```

3544                                     ;TEST ITERATION SUBROUTINE*****
3545
3546 016600 032777 004000 161762 ITER: BIT @4000,@SWR ;SEE IF ITERATIONS
3547 016606 001403 BEQ 2$ ;IF SO: BR
3548 016610 005037 000676 1$: CLR ITCNT ;CLEAR ITERATION COUNTER
3549 016614 000207 RTS PC ;ELSE EXIT
3550 016616 005737 001014 2$: TST PCNTR ;NO SUBTEST ITERATIONS ON FIRST PASS
3551 016622 001772 BEQ 1$
3552 016624 005237 000676 INC ITCNT ;BUMP COUNTER
3553 016630 023737 000676 000602 CMP ITCNT,ITAMT ;SEE IF DONE ALL
3554 016636 001764 BEQ 1$ ;IF SO: BR
3555 016640 005726 TST (SP). ;RESET STACK
3556 016642 017700 162042 MOV @ITRLP,RO ;SET ITERATION POINTER
3557 016646 000110 JMP (RO) ;GO ITERATE
3558
3559                                     ;MANUAL INTERVENTION INHIBIT*****
3560
3561 016650 000240 INMT: NOP
3562 016652 012704 022617 MOV @MSG43,R4
3563 016656 004737 017760 JSR PC,TTOUT ;GO PRINT INHIB MSG
3564 016662 000000 HALT
3565 016664 000137 002530 JMP TSCD2 ;RETURN TO SCHED
3566
3567                                     ;NON-STANDARD MODE TEST HANDLER
3568
3569
3570 016670 010046 NOST: MOV RO,-(SP) ;SAVE RO
3571 016672 012700 000240 MOV @240,RO ;SET UP INDEX
3572 016676 013760 001326 001056 MOV TADX,TSTTBL(RO) ;
3573 016704 005720 TST (RO).
3574 016706 012737 000047 001330 MOV @47,TLAST ;SET END OF TEST
3575 016714 013760 001330 001056 MOV TLAST,TSTTBL(RO) ;SET LAST TEST NUMBER
3576 016722 012600 MOV (SP),RO ;RESTORE RO
3577 016724 000207 RTS PC ;RETURN
3578

```

```

3580
3581 ;INITIALIZE SUBROUTINE*****
3582
3583 016726 000240 INIT1: NOP
3584 016730 012777 000040 161562 MOV #40,@CS ;INIT
3585 016736 013777 000620 161554 INIT2: MOV DRVN,@CS ;SELECT DRIVE
3586 016744 013777 000660 161570 MOV SLVN,@TC ;SELECT SLAVE
3587 016752 000207 RTS PC ;RETURN
3588
3589 ;ROUTINES TO INITIALIZE SLAVE. THESE ROUTINES PLACE THE SLAVE
3590 ;IN PROPER STATUS FOR THE CALLING TEST. INIT3 PLACES THE SLAVE IN
3591 ;NRZ MODE AND OFF BOT; INIT4 PLACES THE SLAVE IN PE MODE AND OFF
3592 ;BOT. IF THE SLAVE IS IN THE PROPER STATUS ON ENTRY NO ACTION IS TAKEN.
3593
3594 ;SET SLAVE IN NRZ OFF BOT
3595 016754 013746 000772 INIT3: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
3596 016760 012737 001400 000772 MOV #1400,UDES ;SET UNIT DESCRIPTION = NRZ
3597 016766 000410 BR INITS ;GO TO INITS ROUTINE
3598
3599 ;SET SLAVE IN PE OFF BOT
3600 016770 013746 000772 INIT4: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
3601 016774 012737 002000 000772 MOV #2000,UDES ;SET UNIT DESCRIPTION = PE
3602 017002 000402 BR INITS ;GO DO IT
3603
3604 ;THIS ROUTINE IS ENTERED AT INIT WHEN THE CALLER HAS SETUP UDES.
3605 ;IT IS ENTERED AT INITS WHEN EITHER INIT3 OR INIT4 HAS SET UP UDES.
3606 017004 013746 000772 INIT: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
3607 017010 012777 000040 161502 INITS: MOV #40,@CS ;INIT CONTROLLER
3608 017016 013777 000620 161474 MOV DRVN,@CS ;SELECT TMO3 DRIVE
3609 017024 013777 000660 161510 MOV SLVN,@TC ;SELECT TE16 SLAVE
3610 017032 013746 000772 MOV UDES,-(SP) ;GET SLAVE DESCRIPTION
3611 017036 042716 174377 BIC #174377,(SP) ;CLEAR ALL BUT DENSITY SELECT BITS
3612 017042 022726 001400 CMP #1400,(SP) ;BRANCH IF REQUESTING PE MODE
3613 017046 001005 BNE 1$
3614 017050 032777 000040 161444 BIT #40,@DS ;BRANCH IF SLAVE IS IN NRZ MODE
3615 017056 001420 BEQ 4$ ;(PES = 0)
3616 017060 000404 BR 2$
3617 017062 032777 000040 161432 1$: BIT #40,@DS ;BRANCH IF SLAVE IS IN PE MODE
3618 017070 001013 BNE 4$
3619 017072 012777 000007 161410 2$: MOV #7,@C1 ;REWIND SLAVE
3620 017100 032777 000200 161414 20$: BIT #200,@DS ;WAIT FOR READY
3621 017106 001774 BEQ 20$
3622 017110 032777 020000 161404 3$: BIT #20000,@DS ;WAIT UNTIL PIP CLEARS
3623 017116 001374 BNE 3$
3624 017120 053777 000772 161414 4$: BIS UDES,@TC ;LOAD SLAVE DESCRIPTION
3625 017126 032777 000002 161366 BIT #2,@DS ;BRANCH IF NOT AT BOT
3626 017134 001407 BEQ 6$
3627 017136 012777 000025 161344 MOV #25,@C1 ;ERASE TO GET OFF BOT
3628 017144 032777 000200 161350 5$: BIT #200,@DS ;WAIT FOR READY
3629 017152 001774 BEQ 5$
3630 017154 032777 000020 161340 6$: BIT #20,@DS ;WAIT FOR SETTLEDOWN TO CLEAR
3631 017162 001374 BNE 6$
3632 017164 012777 000011 161316 MOV #11,@C1 ;RESET DRIVE
3633 017172 012637 000772 MOV (SP)+,UDES ;RESTORE UNIT DESCRIPTION
3634 017176 000207 RTS PC ;RETURN
3635

```



```

3636
3637
3638                                     ;MANUAL INSTRUCTION SUBROUTINE*****
3639
3640 017200 000240          INST:  NOP
3641 017202 004737 017760      JSR    PC,TTOUT          ;PRINT INSTRUCTION
3642 017206 012704 027122      MOV    @MMSG0,R4
3643 017212 004737 017760      JSR    PC,TTOUT          ;PRINT REPLY
3644 017216 012705 000674      MOV    @TEMP3,R5
3645 017222 012701 000001      MOV    #1,R1
3646 017226 012702 177777      MOV    #-1,R2
3647 017232 012703 000000      MOV    #0,R3
3648 017236 004737 017436      JSR    PC,TTR           ;AWAIT REPLY
3649 017242 000240          NOP
3650 017244 000207          RTS    PC                ;EXIT
3651
3652                                     ;MAG TAPE INTERRUPT HANDLER*****
3653
3654 017246 000240          MTINT: NOP
3655 017250 013716 000664      MOV    RTRN,(SP)       ;SET RETURN FROM INTERUPT ADDRESS
3656 017254 000002          RTI                    ;RETURN
3657
3658                                     ;TTY INTERRUPT HANDLER*****
3659
3660 017256 017746 161312      TTINT: MOV    @TKB,-(SP)     ;GET CHARACTER
3661 017262 042716 000200      BIC    @200,(SP)       ;CLEAR PARITY BIT
3662 017266 122716 000003      CMPB   @3,(SP)         ;BRANCH IF NOT CONTROL C
3663 017272 001010          BNE    1$
3664 017274 005737 001422      TST    CHNFLG          ;INHIBIT +C IF IN CHAIN MODE
3665 017300 001005          BNE    1$
3666 017302 005077 161260      CLR    @PSW            ;CLEAR PSW
3667 017306 000005          RESET
3668 017310 000137 000200      JMP    @#200           ;RESTART
3669 017314 122716 000001      1$:  CMPB   @1,(SP)       ;BRANCH IF NOT +A
3670 017320 001017          BNE    2$
3671 017322 022737 000176 000570  CMP    @SWREG,SWR      ;BRANCH IF USING HARWARE SWR
3672 017330 001016          BNE    3$
3673 017332 012737 177570 000570  MOV    @177570,SWR     ;INVOKE HARDWARE SWR
3674 017340 004737 020716      JSR    PC,.SAVE        ;SAVE REGISTERS ON THE STACK
3675 017344 012704 023614      MOV    @MSG63,R4       ;TYPE 'HARDWARE SWR IN USE'
3676 017350 004737 017760      JSR    PC,TTOUT
3677 017354 004737 020740      JSR    PC,.RESTORE
3678 017360 122716 000007      2$:  CMPB   @7,(SP)       ;BRANCH IF NOT +G
3679 017364 001006          BNE    4$

```


3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750

017436 010146
017440 011601
017442 005037 000670
017446 005000
017450 004737 017716
017454 122737 000003 000612
017462 001003
017464 000005
017466 000137 000200
017472 122737 000015 000612
017500 001004
017502 005737 000670
017506 001471
017510 000457
017512 122737 000025 000612
017520 001005
017522 012704 023542
017526 004737 017760
017532 000742
017534 122737 000177 000612
017542 001012
017544 000241
017546 006000
017550 006200
017552 006200
017554 012704 023544
017560 004737 017760
017564 005201
017566 000730
017570 122737 000060 000612
017576 101402
017600 000137 017676
017604 122737 000070 000612
017612 101002
017614 000137 017676
017620 005237 000670
017624 006300
017626 006300
017630 006300

TTR:
10\$:

1\$:

11\$:

2\$:

21\$:

3\$:

4\$:

5\$:

```
*****  
;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  
*****  
MOV R1, -(SP) ;SAVE CHARACTER COUNT  
MOV (SP), R1 ;RESTORE CHARACTER COUNT (FOR +U)  
CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG  
CLR R0  
JSR PC, TTIN ;GO READ CHARACTER  
CMPB #3, TIB ;BRANCH IF NOT +C  
BNE 11$  
RESET ;RESET  
JMP @#200 ;RESTART PROGRAM  
CMPB #15, TIB ;SEE IF CR  
BNE 2$ ;IF NOT: BR  
TST TEMP1 ;SEE IF FIRST CHARACTER  
BEQ 9$ ;IF SO: BR  
BR 6$  
CMPB #25, TIB ;BRANCH IF NOT CONTROL U  
BNE 21$  
MOV #MSG59, R4 ;TYPE <CR><LF>  
JSR PC, TTOUT  
BR 10$  
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$  
CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0  
BLOS 4$ ;IF NOT: BR  
JMP TINNER ;ELSE GO TO ERROR  
CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7  
BHI 5$ ;IF NOT: BR  
JMP TINNER ;ELSE GO TO ERROR  
INC TEMP1 ;SET FIRST CHARACTER FLAG  
ASL R0  
ASL R0 ;SHIFT 3 LEFT  
ASL R0
```

3751	017632	042737	177770	000612		BIC	#177770,TIB	:STRIP ASCII
3752	017640	053700	000612			BIS	TIB,R0	:LOAD CHARACTER
3753	017644	005301				DEC	R1	:SEE IF DONE
3754	017646	001300				BNE	1\$:IF NOT: BR
3755	017650	020002			6\$:	CMP	R0,R2	:SEE IF EXCEEDED MAXIMUM LIMIT
3756	017652	101402				BLOS	7\$:IF NOT: BR
3757	017654	000137	017676			JMP	TINER	:ELSE GO TO ERROR
3758	017660	020300			7\$:	CMP	R3,R0	:SEE IF BELOW MINIMUM LIMIT
3759	017662	101402				BLOS	8\$:IF NOT: BR
3760	017664	000137	017676			JMP	TINER	:ELSE GO TO ERROR
3761	017670	010015			8\$:	MOV	R0,(R5)	:LOAD VALUE
3762	017672	005726			9\$:	TST	(SP).	:POP CHAR COUNT OFF STACK
3763	017674	000207				RTS	PC	:EXIT
3764								


```

3766
3767
3768
3769 017676 012704 022557
3770 017702 004737 017760
3771 017706 005726
3772 017710 162716 000020
3773 017714 000207
3774
3775
3776
3777 017716 005277 160650
3778 017722 105777 160644
3779 017726 100375
3780 017730 017737 160640 000612
3781 017736 042737 000200 000612
3782 017744 013737 000612 000610
3783 017752 004737 020060
3784 017756 000207
3785
3786
3787
3788 017760 112437 000610
3789 017764 122737 000043 000610
3790 017772 001472
3791 017774 122737 000045 000610
3792 020002 001403
3793 020004 004737 020060
3794 020010 000763
3795 020012 112737 000015 000610
3796 020020 004737 020060
3797 020024 012703 000004
3798 020030 005037 000610
3799 020034 004737 020060
3800 020040 005303
3801 020042 001372
3802 020044 112737 000012 000610
3803 020052 004737 020060
3804 020056 000740

;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: INC @TKS
1$: TSJB @TKS
    BPL 1$
    MOV @TKB,TIB ;STRIP PARITY BIT
    BIC #200,TIB ;MOVE CHAR TO TTY OUTPUT BFR
    MOV TIB,TOB ;AND ECHO IT
    JSR PC,TOG
    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

```

```

3806
3807 020060 105777 160506          TOG:  TSTB  @TKS          ;SEE IF INPUT AT KEYBOARD
3808 020064 100024                    BPL  3$          ;IF SO, THEN
3809 020066 117737 160502 001333    MOVB  @TKB,%CNTRLS+1 ;MOVE CHARACTER AND
3810 020074 142737 000200 001333    BICB  @200,%CNTRLS+1 ;MASK OFF PARITY BIT.
3811 020102 122737 000023 001333    CMPB  @23,%CNTRLS+1 ;SEE IF CHARACTER IS XOFF
3812 020110 001004                    BNE  2$          ;IF XOFF, THEN
3813 020112 112737 000377 001332    MOVB  @377,%CNTRLS  ;SET XOFF FLAG
3814 020120 000757                    BR   TOG
3815 020122 122737 000021 001333  2$:  CMPB  @21,%CNTRLS+1 ;SEE IF CHARACTER IS XON
3816 020130 001002                    BNE  3$          ;IF SO THEN
3817 020132 105037 001332          CLRB  %CNTRLS    ;CLEAR XOFF FLAG
3818 020136 105737 001332          3$:  TSTB  %CNTRLS  ;SEE IF IN XOFF MODE
3819 020142 100746                    BMI  TOG         ;IF NOT THEN
3820 020144 105777 160426          TSTB  @TPS      ;CHECK IF PRINTER READY
3821 020150 100343                    BPL  TOG
3822 020152 113777 000610 160420    MOVB  TOB,@TPB
3823 020160 000207          TEX:  RTS      PC          ;RETURN

```

;OCTAL OUTPUT SUBROUTINE*****

```

3824
3825
3826
3827
3828 020162 012737 000001 020412  OCTPE:  MOV   #1,OFL
3829 020170 000402                    BR   OCTPE1
3830 020172 005037 020412          OCTP:  CLR   OFL          ;CLEAR FLAG FOR LEADING ZERO
3831 020176 010304                    OCTPE1: MOV  R3,R4        ;SEE IF NUMBER IS ZERO
3832 020200 001006                    BNE  OCTP0          ;IF NOT ZERO: BR
3833 020202 005737 020412          TST   OFL          ;SEE IF PRINT ALL 0
3834 020206 001003                    BNE  OCTP0          ;IF SO: BR
3835 020210 004737 020372          JSR   PC,OCTPG1    ;ELSE PRINT ZERO
3836 020214 000447                    BR   OCTP3          ;SPACE AND EXIT
3837 020216 032704 100000          OCTP0: BIT  @100000,R4 ;SEE IF MSD = 1
3838 020222 001405                    BEQ  OCTP1          ;IF NOT: BR
3839 020224 012704 000001          MOV   #1,R4
3840 020230 004737 020350          JSR   PC,OCTPG    ;PRINT 1
3841 020234 000403                    BR   OCTP2
3842 020236 005004                    OCTP1: CLR  R4
3843 020240 004737 020350          JSR   PC,OCTPG    ;PRINT 0
3844 020244 010304                    OCTP2: MOV  R3,R4
3845 020246 006004                    ROR  R4
3846 020250 006004                    ROR  R4
3847 020252 006004                    ROR  R4          ;POSITION DIGIT
3848 020254 006004                    ROR  R4
3849 020256 000304                    SWAB R4
3850 020260 004737 020350          JSR   PC,OCTPG    ;PRINT DIGIT 2
3851 020264 010304                    MOV  R3,R4
3852 020266 006004                    ROR  R4
3853 020270 000304                    SWAB R4
3854 020272 004737 020350          JSR   PC,OCTPG    ;PRINT DIGIT 3
3855 020276 010304                    MOV  R3,R4
3856 020300 006104                    ROL  R4
3857 020302 006104                    ROL  R4
3858 020304 000304                    SWAB R4
3859 020306 004737 020350          JSR   PC,OCTPG    ;PRINT DIGIT 4
3860 020312 010304                    MOV  R3,R4
3861 020314 006004                    ROR  R4

```


3862	020316	006004			ROR	R4	
3863	020320	006004			ROR	R4	
3864	020322	004737	020350		JSR	PC,OCTPG	
3865	020326	010304			MOV	R3,R4	
3866	020330	004737	020350		JSR	PC,OCTPG	;PRINT DIGIT 5
3867	020334	012737	000240	000610	MOV	#240,TOB	
3868	020342	004737	020060		JSR	PC,TOG	;PRINT SPACE
3869	020346	000207			RTS	PC	;EXIT
3870							
3871	020350	042704	177770		OCTPG:	BIC	#177770,R4
3872	020354	001004			BNE	OCTPGO	
3873	020356	005737	020412		TST	OFL	
3874	020362	001001			BNE	OCTPGO	
3875	020364	000207			RTS	PC	
3876							
3877	020366	005237	020412		OCTPGO:	INC	OFL
3878	020372	052704	000260		OCTPG1:	BIS	#260,R4
3879	020376	010437	000610		MOV	R4,TOB	
3880	020402	004737	020060		JSR	PC,TOG	
3881	020406	010304			MOV	R3,R4	
3882	020410	000207			RTS	PC	
3883	020412	000000			OFL:	0	;FIRST CHAR FLAG
3884							

```

3886
3887
3888 ;DATA CHARACTER OUTPUT SUBROUTINE*****
3889 020414 012704 000010 DOUT: MOV #10,R4 ;SET NUMBER TO PRINT
3890 020420 110337 000610 MOVB R3,TOB
3891 020424 105777 160146 1$: TSTB @TPS
3892 020430 100375 BPL 1$
3893 020432 132737 000200 000610 BITB #200,TOB
3894 020440 001404 BEQ 2$
3895 020442 012777 000061 160130 MOV #061,@TPB
3896 020450 000403 BR 3$
3897 020452 012777 000060 160120 2$: MOV #060,@TPB
3898 020460 006337 000610 3$: ASL TOB
3899 020464 005304 DEC R4
3900 020466 001356 BNE 1$
3901 020470 000207 RTS PC
3902
3903 020472 013703 000674 DOUTD: MOV TEMP3,R3
3904 020476 000303 SWAB R3
3905 020500 004737 020414 JSR PC,DOUT
3906 020504 013703 000674 MOV TEMP3,R3
3907 020510 004737 020414 JSR PC,DOUT
3908 020514 000207 RTS PC
3909
3910 ;TE16 SERIAL NUMBER PRINT SUBROUTINE*****
3911
3912 020516 010304 SNPT: MOV R3,R4
3913 020520 000304 SWAB R4
3914 020522 006004 ROR R4
3915 020524 006004 ROR R4
3916 020526 006004 ROR R4
3917 020530 006004 ROR R4 ;GET FIRST DIGIT
3918 020532 004737 020574 JSR PC,SNPG ;PRINT
3919 020536 010304 MOV R3,R4
3920 020540 000304 SWAB R4 ;GET SECOND DIGIT
3921 020542 004737 020574 JSR PC,SNPG ;PRINT
3922 020546 010304 MOV R3,R4
3923 020550 006004 ROR R4
3924 020552 006004 ROR R4
3925 020554 006004 ROR R4
3926 020556 006004 ROR R4
3927 020560 004737 020574 JSR PC,SNPG ;PRINT THIRD DIGIT
3928 020564 010304 MOV R3,R4
3929 020566 004737 020574 JSR PC,SNPG ;PRINT FOURTH DIGIT
3930 020572 000207 RTS PC ;EXIT
3931 020574 012737 000260 000610 SNPG: MOV #260,TOB ;SET BASE = 0
3932 020602 042704 177760 BIC #177760,R4 ;MASK DIGIT
3933 020606 050437 000610 BIS R4,TOB ;SET ASCII
3934 020612 004737 020060 JSR PC,TOG ;TYPE DIGIT
3935 020616 000207 RTS PC ;RETURN

```



```

3937
3938
3939
3940 020620 022737 000176 00057C GTSWR: CMP #SWREG,SWR ;ROUTINE TO LOAD CONTENTS OF SOFTWARE SWITCH REGISTER.
3941 020626 001032 BNE 1$ ;IF A CONTROL G (+G) IS TYPED THE SOFTWARE SWITCH REGISTER IS LOADED
3942 020630 004737 020716 JSR PC,.SAVE ;BRANCH IF SOFTWARE SWR
3943 020634 012704 027473 MOV #MSWR,R4 ;NOT INVOKED
3944 020640 004737 017760 JSR PC,TTOUT ;SAVE REGISTERS ON THE STACK
3945 020644 017703 157720 MOV @SWR,R3 ;TYPE 'SWR = '
3946 020650 004737 020162 JSR PC,OCTPE ;GET CURRENT VALUE
3947 020654 012704 027503 MOV #MNEW,R4 ;AND TYPE IT
3948 020660 004737 017760 JSR PC,TTOUT ;ASK FOR NEW VALUE
3949 020664 013705 000570 MOV SWR,R5 ;NEW VALUE WILL BE RETURNED IN (R5)
3950 020670 012701 000007 MOV #7,R1 ;LIMIT TO 7 CHARACTERS
3951 020674 012702 177777 MOV #177777,R2 ;LIMIT RESPONSE TO BETWEEN
3952 020700 012703 000000 MOV #0,R3 ;0 AND 177777
3953 020704 004737 017436 JSR PC,TTR ;GET RESPONSE
3954 020710 004737 020740 JSR PC,.RESTORE ;RESTORE REGISTERS
3955 020714 000207 1$: RTS PC ;RETURN TO CALLER
3956
;;ROUTINE TO SAVE REGISTERS ON THE STACK
.SAVE: MOV #5,-(SP) ;;R5 IS SAVED AT 12(SP)
MOV #4,-(SP) ;;R4 IS SAVED AT 10(SP)
MOV #3,-(SP) ;;R3 IS SAVED AT 6(SP)
MOV #2,-(SP) ;;R2 IS SAVED AT 4(SP)
MOV #1,-(SP) ;;R1 IS SAVED AT 2(SP)
MOV #0,-(SP) ;;R0 IS SAVED AT (SP)
MOV 14(SP),-(SP) ;;PUSH RETURN PC ON THE STACK
RTS PC ;;RETURN TO CALLER
3957
;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
.RESTORE:MOV (SP)+,14(SP) ;;STORE RETURN PC ON STACK
MOV (SP)+,#0
MOV (SP)+,#1
MOV (SP)+,#2
MOV (SP)+,#3
MOV (SP)+,#4
MOV (SP)+,#5
RTS PC ;;RETURN
3958
3959 ;MESSAGE TABLE*****
3960
3961 020762 022445 046524 031460 MSG1: .ASCII 'TM03-TE16/TU77 CONTROL LOGIC TEST- PART I (CZTEAEO)';.B
020770 052055 030505 027466
020776 052524 033467 041440
021004 047117 051124 046117
021012 046040 043517 041511
021020 052040 051505 026524
021026 050040 051101 020124
021034 020111 041450 052132
021042 040505 030105 051
3962 021047 045 025052 040452 .ASCII /*****ASSURE TAPE IS AT BOT****/
021054 051523 051125 020105
021062 040524 042520 044440
021070 020123 052101 041040
021076 052117 025052 052

```

3963	021103	045	054524	042520		.ASCII	/TYPE <CR> TO TERMINATE RESPONSE, & +C TO RESTART##/
	021110	036040	051103	020076			
	021116	047524	052040	051105			
	021124	044515	040516	042524			
	021132	051040	051505	047520			
	021140	051516	020105	020046			
	021146	041536	052040	020117			
	021154	042522	052123	051101			
	021162	022524	043				
3964	021165	045	051104	053111	MSG2:	.ASCII	/DRIVE NUMBER OR (CR) WHEN DONE #/
	021172	020105	052516	041115			
	021200	051105	047440	020122			
	021206	041450	024522	053440			
	021214	042510	020116	047504			
	021222	042516	021440				
3965	021226	022445	047506	020122	MSG2A:	.ASCII	/FOR DRIVE ADDRESS TEST;/
	021234	051104	053111	020105			
	021242	042101	051104	051505			
	021250	020123	042524	052123			
	021256	073					
3966	021257	045	042440	052116		.ASCII	/ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
	021264	051105	042440	050130			
	021272	020124	051104	053111			
	021300	020105	052516	041115			
	021306	051105	020054	046101			
	021314	020114	052117	042510			
	021322	051522	051440	047510			
	021330	046125	020104	042502			
	021336	047040	047117	042455			
	021344	044530	052123	047101			
	021352	027124	043				
3967	021355	045	047516	026516	MSG3:	.ASCII	/NON-EXIST DRIVE #/
	021362	054105	051511	020124			
	021370	051104	053111	020105			
	021376	043					
3968	021377	045	044122	042040	MSG4:	.ASCII	/RH DETECTED #/
	021404	052105	041505	042524			
	021412	020104	043				
3969	021415	045	046524	031460	MSG5:	.ASCII	/TM03 DETECTED #/
	021422	042040	052105	041505			
	021430	042524	020104	043			
3970	021435	105	050130	026524	MSG6:	.ASCII	/EXPT-NOT RECVD#/
	021442	047516	020124	042522			
	021450	053103	021504				
3971	021454	041522	042126	047055	MSG7:	.ASCII	/RCVD-NOT EXPT#/
	021462	052117	042440	050130			
	021470	021524					
3972	021472	051445	040514	042526	MSG8:	.ASCII	/SLAVE NUMBER OR (CR) WHEN DONE #/
	021500	047040	046525	042502			
	021506	020122	051117	024040			
	021514	051103	020051	044127			
	021522	047105	042040	047117			
	021530	020105	043				
3973	021533	045	043045	051117	MSG8A:	.ASCII	/FOR SLAVE ADDRESS TEST;/
	021540	051440	040514	042526			
	021546	040440	042104	042522			

	021554	051523	052040	051505		
	021562	035524				
3974	021564	020045	047105	042524	.ASCII	/# ENTER EXPT SLAVE NUMBER, ALL OTHERS SHOULD BE NON EXISTANT.#/
	021572	020122	054105	052120		
	021600	051440	040514	042526		
	021606	047040	046525	042502		
	021614	026122	040440	046114		
	021622	047440	044124	051105		
	021630	020123	044123	052517		
	021636	042114	041040	020105		
	021644	047516	026516	054105		
	021652	051511	040524	052116		
	021660	021456				
3975	021662	047045	047117	042455	MSG9:	.ASCII /#NON-EXIST SLAVE #/
	021670	044530	052123	051440		
	021676	040514	042526	021440		
3976	021704	051045	040505	020104	MSG10:	.ASCII /#READ CONT BUS PAR #/
	021712	047503	052116	041040		
	021720	051525	050040	051101		
	021726	021440				
3977	021730	053445	044522	042524	MSG11:	.ASCII /#WRITE CONT BUS PAR #/
	021736	041440	047117	020124		
	021744	052502	020123	040520		
	021752	020122	043			
3978	021755	040	054105	052120	MSG12:	.ASCII / EXPT #/
	021762	021440				
3979	021764	051040	053103	020104	MSG13:	.ASCII / RCVD #/
	021772	043				
3980	021773	045	051115	041040	MSG14:	.ASCII /#MR BITS 4-0#/
	022000	052111	020123	026464		
	022006	021460				
3981	022010	046445	020122	044502	MSG15:	.ASCII /#MR BITS 15-7#/
	022016	051524	030440	026465		
	022024	021467				
3982	022026	044445	042524	035122	MSG16:	.ASCII /#ITER: #/
	022034	021440				
3983	022036	052045	020103	044502	MSG18:	.ASCII /#TC BITS 12-0 #/
	022044	051524	030440	026462		
	022052	020060	043			
3984	022055	045	041506	041040	MSG19:	.ASCII /#FC BITS 15-0 #/
	022062	052111	020123	032461		
	022070	030055	021440			
3985	022074	043045	047125	041440	MSG20:	.ASCII /#FUN CODE BITS 5-1 OF C1 #/
	022102	042117	020105	044502		
	022110	051524	032440	030455		
	022116	047440	020106	030503		
	022124	021440				
3986	022126	043445	020117	044502	MSG21:	.ASCII /#GO BIT NOT CORRECT AT START #/
	022134	020124	047516	020124		
	022142	047503	051122	041505		
	022150	020124	052101	051440		
	022156	040524	052122	021440		
3987	022164	043445	020117	044502	MSG22:	.ASCII /#GO BIT NOT SET #/
	022172	020124	047516	020124		
	022200	042523	020124	043		
3988	022205	045	047507	041040	MSG23:	.ASCII /#GO BIT NOT RESET BY INIT #/

	022212	052111	047040	052117			
	022220	051040	051505	052105			
	022226	041040	020131	047111			
3989	022234	052111	021440				
	022240	042045	054522	047040	MSG24:	.ASCII	/DRY NOT SET BY INIT #/
	022246	052117	051440	052105			
	022254	041040	020131	047111			
3990	022262	052111	021440				
	022266	042045	054522	047040	MSG25:	.ASCII	/DRY NOT RESET BY GO-1#/
	022274	052117	051040	051505			
	022302	052105	041040	020131			
	022310	047507	030475	043			
3991	022315	045	051104	020131	MSG25A:	.ASCII	/DRY NOT SET BY GO-0#/
	022322	047516	020124	042523			
	022330	020124	054502	043440			
3992	022336	036517	021460				
	022342	047045	020117	047111	MSG26:	.ASCII	/NO INTERRUPT RETURNED#/
	022350	042524	051122	050125			
	022356	020124	042522	052524			
	022364	047122	042105	043			
3993	022371	045	040502	020104	MSG27:	.ASCII	/BAD STATUS#/
	022376	052123	052101	051525			
	022404	043					
3994	022405	040	047123	020072	MSG30:	.ASCII	/SN: #/
	022412	043					
3995	022413	045	051105	020122	MSG31:	.ASCII	/ERR NOT SET #/
	022420	047516	020124	042523			
	022426	020124	043				
3996	022431	045	052101	020101	MSG32:	.ASCII	/ATA NOT SET #/
	022436	047516	020124	042523			
	022444	020124	043				
3997	022447	045	051501	041040	MSG33:	.ASCII	/AS BIT NOT SET #/
	022454	052111	047040	052117			
	022462	051440	052105	021440			
3998	022470	051445	020103	047516	MSG34:	.ASCII	/SC NOT SET #/
	022476	020124	042523	020124			
	022504	043					
3999	022505	045	051124	020105	MSG35:	.ASCII	/TRE NOT SET #/
	022512	047516	020124	042523			
	022520	020124	043				
4000	022523	045	046123	020101	MSG36:	.ASCII	/SLA NOT SET #/
	022530	047516	020124	042523			
	022536	020124	043				
4001	022541	045	051523	020103	MSG37:	.ASCII	/SSC NOT SET #/
	022546	047516	020124	042523			
	022554	020124	043				
4002	022557	040	020077	043	MSG40:	.ASCII	/ ? #/
4003	022563	045	042445	042116	MSG41:	.ASCII	/END OF PASS #/
	022570	047440	020106	040520			
	022576	051523	021440				
4004	022602	042045	040505	020104	MSG42:	.ASCII	/DEAD TRACK #/
	022610	051124	041501	020113			
	022616	043					
4005	022617	045	046445	047101	MSG43:	.ASCII	/MANUAL TESTS (14 17) INHIBITED: HALT #/
	022624	040525	020114	042524			
	022632	052123	020123	030450			


```

4017 023313 045 052123 052101 MSG56: .ASCII /#STATIC TESTS ONLY: #/
      023320 041511 052040 051505
      023326 051524 047440 046116
      023334 035131 021440
4018 023340 052045 047515 020063 MSG57: .ASCII /#TM03 DRIVE: #/
      023346 051104 053111 035105
      023354 021440
4019 023356 044445 020123 047503 MS57A: .ASCII /#IS CONTROLLER JUMPERED IN NON-STANDARD MODE./<15><12>
      023364 052116 047522 046114
      023372 051105 045040 046525
      023400 042520 042522 020104
      023406 047111 047040 047117
      023414 051455 040524 042116
      023422 051101 020104 047515
      023430 042504 006454 012
4020 023435 124 050131 020105 .ASCII /TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD ? #/
      023442 020062 047506 020122
      023450 047516 036516 052123
      023456 047101 040504 042122
      023464 047440 020122 051103
      023472 043040 051117 051440
      023500 040524 042116 051101
      023506 020104 020077 020040
      023514 020040 043
4021 023517 045 042524 033061 MSG58: .ASCII /#TE16/TU77 SLAVE: #/ ;..B
      023524 052057 033525 020067
      023532 046123 053101 035105
      023540 021440
4022 023542 021445 MSG59: .ASCII /#0/
4023 023544 021534 MSG60: .ASCII /\0/
4024 023546 051045 046505 053117 MSG62: .ASCII /#REMOVE TM0P FROM SLAVE TO BE TESTED##/
      023554 020105 046524 050104
      023562 043040 047522 020115
      023570 046123 053101 020105
      023576 047524 041040 020105
      023604 042524 052123 042105
      023612 021445
4025 023614 044045 051101 053504 MSG63: .ASCII /#HARDWARE SWR IN USE##/
      023622 051101 020105 053523
      023630 020122 047111 052440
      023636 042523 021445
4026 023642 051445 040514 042526 MSG64: .ASCII /#SLAVE TYPE: #/ ;..B
      023650 052040 050131 035105
      023656 021440
4027 023660 052524 033467 043 MSG65: .ASCII /TU77#/ ;..B
4028 023665 124 030505 021466 MSG66: .ASCII /TE16#/ ;..B
4029 023672 051445 040514 042526 MSG67: .ASCII /#SLAVE TYPE (0=TE16,1=TU77): #/ ;..B
      023700 052040 050131 020105
      023706 030050 052075 030505
      023714 026066 036461 052524
      023722 033467 035051 021440
4030 023730 022445 047111 047503 MSG68: .ASCII /#INCORRECT SLAVE TYPE!!! PROGRAM ABORTED#/ ;..B
      023736 051122 041505 020124
      023744 046123 053101 020105
      023752 054524 042520 020441
      023760 020041 051120 043517

```


	023766	040522	020115	041101		
	023774	051117	042524	021504		
4031	024002	046111	042514	040507	MSG69:	.ASCII /ILLEGAL#/ ;..B
	024010	021514				
4032						;TEST HEADER*****
4033						
4034	024012	022445	047514	044507	MSLT1:	.ASCII /LOGIC TEST 1: DRIVE ADDRESSING (M8909 RH)#/
	024020	020103	042524	052123		
	024026	030440	020072	051104		
	024034	053111	020105	042101		
	024042	051104	051505	044523		
	024050	043516	024040	034115		
	024056	030071	020071	044122		
	024064	021451				
4035	024066	022445	047514	044507	MSLT2:	.ASCII /LOGIC TEST 2: REGISTER ADDRESSING (M8909 RH)#/
	024074	020103	042524	052123		
	024102	031040	020072	042522		
	024110	044507	052123	051105		
	024116	040440	042104	042522		
	024124	051523	047111	020107		
	024132	046450	034470	034460		
	024140	051040	024510	043		
4036	024145	045	046045	043517	MSLT3:	.ASCII /LOGIC TEST 3: CONTROL BUS TEST (RH M8905-YB M8909)#/
	024152	041511	052040	051505		
	024160	020124	035063	041440		
	024166	047117	051124	046117		
	024174	041040	051525	052040		
	024202	051505	020124	051050		
	024210	020110	034115	030071		
	024216	026465	041131	046440		
	024224	034470	034460	021451		
4037	024232	022445	047514	044507	MSLT4:	.ASCII /LOGIC TEST 4: SLAVE ADDRESSING (M8905-YB M8933)#/
	024240	020103	042524	052123		
	024246	032040	020072	046123		
	024254	053101	020105	042101		
	024262	051104	051505	044523		
	024270	043516	024040	034115		
	024276	030071	026465	041131		
	024304	046440	034470	031463		
	024312	021451				
4038	024314	022445	047514	044507	MSLT5:	.ASCII /LOGIC TEST 5: MR BIT TEST (M8905-YB)#/
	024322	020103	042524	052123		
	024330	032440	020072	051115		
	024336	041040	052111	052040		
	024344	051505	020124	046450		
	024352	034470	032460	054455		
	024360	024502	043			
4039	024363	045	046045	043517	MSLT6:	.ASCII /LOGIC TEST 6: TC BIT TEST (M8905-YB)#/
	024370	041511	052040	051505		
	024376	020124	035066	052040		
	024404	020103	044502	020124		
	024412	042524	052123	024040		
	024420	034115	030071	026465		
	024426	041131	021451			
4040	024432	022445	047514	044507	MSLT7:	.ASCII /LOGIC TEST 7: FC BIT TEST (M8905-YB)#/
	024440	020103	042524	052123		

	024446	033440	020072	041506	
	024454	041040	052111	052040	
	024462	051505	020124	046450	
	024470	034470	032460	054455	
	024476	024502	043		
4041	024501	045	046045	043517	MSLT10: .ASCII /LOGIC TEST 10: FUNCTION BIT TEST (M8905-1B)*/
	024506	041511	052040	051505	
	024514	020124	030061	020072	
	024522	052506	041516	044524	
	024530	047117	041040	052111	
	024536	052040	051505	020124	
	024544	046450	034470	032460	
	024552	054455	024502	043	
4042	024557	045	046045	043517	MSLT11: .ASCII /LOGIC TEST 11: GO BIT TEST (M8909)*/
	024564	041511	052040	051505	
	024572	020124	030461	020072	
	024600	047507	041040	052111	
	024606	052040	051505	020124	
	024614	046450	034470	034460	
	024622	021451			
4043	024624	022445	047514	044507	MSLT12: .ASCII /LOGIC TEST 12: DRIVE READY BIT (M8909)*/
	024632	020103	042524	052123	
	024640	030440	035062	042040	
	024646	044522	042526	051040	
	024654	040505	054504	041040	
	024662	052111	024040	034115	
	024670	030071	024471	043	
4044	024675	045	046045	043517	MSLT13: .ASCII /LOGIC TEST 13: INTERRUPT TEST (RH)*/
	024702	041511	052040	051505	
	024710	020124	031461	020072	
	024716	047111	042524	051122	
	024724	050125	020124	042524	
	024732	052123	024040	044122	
	024740	021451			
4045	024742	022445	047514	044507	MSLT14: .ASCII /LOGIC TEST 14: STATUS AT BOT,ON LINE,WRITE PROTECTED (NO WRITE RING)*/
	024750	020103	042524	052123	
	024756	030440	035064	051440	
	024764	040524	052524	020123	
	024772	052101	041040	052117	
	025000	047454	026516	044514	
	025006	042516	053454	044522	
	025014	042524	050040	047522	
	025022	042524	052103	042105	
	025030	024040	047516	053440	
	025036	044522	042524	051040	
	025044	047111	024507	043	
4046	025051	045	046045	043517	MSLT15: .ASCII /LOGIC TEST 15: STATUS AT BOT,OFF-LINE,WRITE PROTECTED*/
	025056	041511	052040	051505	
	025064	020124	032461	020072	
	025072	052123	052101	051525	
	025100	040440	020124	047502	
	025106	026124	043117	026506	
	025114	044514	042516	053454	
	025122	044522	042524	050040	
	025130	047522	042524	052103	
	025136	042105	043		

4047	025141	045	046045	043517	MSLT16: .ASCII /LOGIC TEST 16: STATUS AT EOT,ON-LINE,WRITE PROTECTED#
	025146	041511	052040	051505	
	025154	020124	033061	020072	
	025162	052123	052101	051525	
	025170	040440	020124	047505	
	025176	026124	047117	046055	
	025204	047111	026105	051127	
	025212	052111	020105	051120	
	025220	052117	041505	042524	
	025226	021504			
4048	025230	022445	047514	044507	MSLT17: .ASCII /LOGIC TEST 17: STATUS AT ON-LINE,WRITE ENABLED#
	025236	020103	042524	052123	
	025244	030440	035067	051440	
	025252	040524	052524	020123	
	025260	052101	047440	026516	
	025266	044514	042516	053454	
	025274	044522	042524	042440	
	025302	040516	046102	042105	
	025310	043			
4049	025311	045	046045	043517	MSLT20: .ASCII /LOGIC TEST 20: ILLEGAL FUNCTION TEST (M8909)#
	025316	041511	052040	051505	
	025324	020124	030062	020072	
	025332	046111	042514	040507	
	025340	020114	052506	041516	
	025346	044524	047117	052040	
	025354	051505	020124	046450	
	025362	034470	034460	021451	
4050	025370	022445	047514	044507	MSLT21: .ASCII /LOGIC TEST 21: RMR(M8909)#
	025376	020103	042524	052123	
	025404	031040	035061	051040	
	025412	051115	046450	034470	
	025420	034460	021451		
4051	025424	022445	047514	044507	MSLT22: .ASCII /LOGIC TEST 22: CPAR(M8909)#
	025432	020103	042524	052123	
	025440	031040	035062	041440	
	025446	040520	024122	034115	
	025454	030071	024471	043	
4052	025461	045	046045	043517	MSLT23: .ASCII /LOGIC TEST 23: FMT(M8905-1B M8906)#
	025466	041511	052040	051505	
	025474	020124	031462	020072	
	025502	046506	024124	034115	
	025510	030071	026465	041131	
	025516	046440	034470	033060	
	025524	021451			
4053	025526	022445	047514	044507	MSLT24: .ASCII /LOGIC TEST 24: DPAR(M8906 RH)#
	025534	020103	042524	052123	
	025542	031040	035064	042040	
	025550	040520	024122	034115	
	025556	030071	020066	044122	
	025564	021451			
4054	025566	022445	047514	044507	MSLT25: .ASCII /LOGIC TEST 25: NEF(M8909)#
	025574	020103	042524	052123	
	025602	031040	035065	047040	
	025610	043105	046450	034470	
	025616	034460	021451		
4055	025622	022445	047514	044507	MSLT26: .ASCII /LOGIC TEST 26: FCE(M8909)#

	025630	020103	042524	052123	
	025636	031040	035066	043040	
	025644	042503	046450	034470	
	025652	034460	021451		
4056	025656	022445	047514	044507	MSLT27: .ASCII /LOGIC TEST 27: ILR(M8909)#/
	025664	020103	042524	052123	
	025672	031040	035067	044440	
	025700	051114	046450	034470	
	025706	034460	021451		
4057	025712	022445	047514	044507	MSLT30: .ASCII /LOGIC TEST 30:DTE(M8906 RH)#/
	025720	020103	042524	052123	
	025726	031440	035060	052104	
	025734	024105	034115	030071	
	025742	020066	044122	021451	
4058	025750	022445	047514	044507	MSLT31: .ASCII /LOGIC TEST 31: OPI(M8933)#/
	025756	020103	042524	052123	
	025764	031440	035061	047440	
	025772	044520	046450	034470	
	026000	031463	021451		
4059	026004	022445	047514	044507	MSLT32: .ASCII /LOGIC TEST 32: UNS(M8909)#/
	026012	020103	042524	052123	
	026020	031440	035062	052440	
	026026	051516	046450	034470	
	026034	034460	021451		
4060	026040	022445	047514	044507	MSLT33: .ASCII /LOGIC TEST 33: PIP(M8909)#/
	026046	020103	042524	052123	
	026054	031440	035063	050040	
	026062	050111	046450	034470	
	026070	034460	021451		
4061	026074	022445	047514	044507	MSLT34: .ASCII /LOGIC TEST 34: PES(M8931)#/
	026102	020103	042524	052123	
	026110	031440	035064	050040	
	026116	051505	046450	034470	
	026124	030463	021451		
4062	026130	022445	047514	044507	MSLT35: .ASCII /LOGIC TEST 35: SAC(M8933 M8905 -1B)#/
	026136	020103	042524	052123	
	026144	031440	035065	051440	
	026152	041501	046450	034470	
	026160	031463	046440	034470	
	026166	032460	054455	024502	
	026174	043			
4063	026175	045	046045	043517	MSLT36: .ASCII /LOGIC TEST 36: FCS(M8933 M8905 -1B)#/
	026202	041511	052040	051505	
	026210	020124	033063	020072	
	026216	041506	024123	034115	
	026224	031471	020063	034115	
	026232	030071	026465	041131	
	026240	021451			
4064	026242	022445	047514	044507	MSLT37: .ASCII /LOGIC TEST 37: ACCL(M8933 M8905 -1B)#/
	026250	020103	042524	052123	
	026256	031440	035067	040440	
	026264	041503	024114	034115	
	026272	031471	020063	034115	
	026300	030071	026465	041131	
	026306	021451			
4065	026310	022445	047514	044507	MSLT40: .ASCII /LOGIC TEST 40: PE TAPE MARK(M8932)#/

	026316	020103	042524	052123	
	026324	032040	035060	050040	
	026332	020105	040524	042520	
	026340	046440	051101	024113	
	026346	034115	031471	024462	
	026354	043			
4066	026355	045	046045	043517	MSLT41: .ASCII /LOGIC TEST 41: NRZ TAPE MARK (M8934)#/
	026362	041511	052040	051505	
	026370	020124	030464	020072	
	026376	051116	020132	040524	
	026404	042520	046440	051101	
	026412	020113	046450	034470	
	026420	032063	021451		
4067	026424	022445	047514	044507	MSLT42: .ASCII /LOGIC TEST 42: CRC(M8934)#/
	026432	020103	042524	052123	
	026440	032040	035062	041440	
	026446	041522	046450	034470	
	026454	032063	021451		
4068	026460	022445	047514	044507	MSLT43: .ASCII /LOGIC TEST 43: LRC(M8934)#/
	026466	020103	042524	052123	
	026474	032040	035063	046040	
	026502	041522	046450	034470	
	026510	032063	021451		
4069	026514	022445	047514	044507	MSLT44: .ASCII /LOGIC TEST 44: CORRECTABLE DATA (M8932 M8901)#/
	026522	020103	042524	052123	
	026530	032040	035064	041440	
	026536	051117	042522	052103	
	026544	041101	042514	042040	
	026552	052101	020101	046450	
	026560	034470	031063	046440	
	026566	034470	030460	021451	
4070	026574	022445	047514	044507	MSLT45: .ASCII /LOGIC TEST 45: INCORRECTABLE DATA (M8932 M8934)#/
	026602	020103	042524	052123	
	026610	032040	035065	044440	
	026616	041516	051117	042522	
	026624	052103	041101	042514	
	026632	042040	052101	020101	
	026640	046450	034470	031063	
	026646	046440	034470	032063	
	026654	021451			
4071	026656	022445	047514	044507	MSLT46: .ASCII /LOGIC TEST 46: PEF(M8932)#/
	026664	020103	042524	052123	
	026672	032040	035066	050040	
	026700	043105	046450	034470	
	026706	031063	021451		
4072	026712	022445	047514	044507	MSLT47: .ASCII /LOGIC TEST 47: FC OVERFLOW (M8905-YB)#/
	026720	020103	042524	052123	
	026726	032040	035067	043040	
	026734	020103	053117	051105	
	026742	046106	053517	024040	
	026750	034115	030071	026465	
	026756	041131	021451		
4073	026762	022445	047514	044507	MSLT50: .ASCII /LOGIC TEST 50: NEF WHEN WRITE PE ON NRZ SLAVE#/
	026770	020103	042524	052123	
	026776	032440	035060	047040	
	027004	043105	053440	042510	

	027012	020116	051127	052111
	027020	020105	042520	047440
	027026	020116	051116	020132
	027034	046123	053101	021505
4074	027042	022445	047514	044507
	027050	020103	042524	052123
	027056	032440	035061	047040
	027064	043105	053440	042510
	027072	020116	051127	052111
	027100	020105	051116	020132
	027106	047117	050040	020105
	027114	046123	053101	021505

MSLT51: .ASCII /LOGIC TEST 51: NEF WHEN WRITE NRZ ON PE SLAVE#/


```

4076
4077
4078
4079 027122 052045 050131 020105 MMSG0: .ASCII /%TYPE CR WHEN READY;#/
      027130 051103 053440 042510
      027136 020116 042522 042101
      027144 035531 043
4080 027147 045 046445 052517 MMSG1: .ASCII /%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE:#/
      027154 052116 052040 050101
      027162 020105 044527 044124
      027170 047040 020117 051127
      027176 052111 020105 044522
      027204 043516 020054 047514
      027212 042101 052040 020117
      027220 047502 026124 051440
      027226 052105 052040 020117
      027234 047117 046040 047111
      027242 035105 043
4081 027245 045 042523 020124 MMSG2: .ASCII /%SET TO OFFLINE:#/
      027252 047524 047440 043106
      027260 044514 042516 021472
4082 027266 046445 053117 020105 MMSG3: .ASCII /%MOVE FORWARD TO EOT, ONLINE:#/
      027274 047506 053522 051101
      027302 020104 047524 042440
      027310 052117 020054 047117
      027316 044514 042516 021472
4083 027324 052445 046116 040517 MMSG4: .ASCII /%UNLOAD, INSERT WRITE RING, LOAD TO BOT, OFFLINE FORWARD PAST BOT, ON L
      027332 026104 044440 051516
      027340 051105 020124 051127
      027346 052111 020105 044522
      027354 043516 020054 047514
      027362 042101 052040 020117
      027370 047502 026124 047440
      027376 043106 044514 042516
      027404 043040 051117 040527
      027412 042122 050040 051501
      027420 020124 047502 026124
      027426 047440 020116 044514
      027434 042516 043
4084 027437 045 046445 053117 MMSG5: .ASCII /%MOVE TAPE TO BOT; ON LINE#/
      027444 020105 040524 042520
      027452 052040 020117 047502
      027460 035524 047440 020116
      027466 044514 042516 043

```

```

4086
4087
4088
4089 027473 045 053523 020122 $MSWR: .ASCII /%SWR = #/
      027500 020075 043
4090 027503 040 042516 020127 $MNEW: .ASCII / NEW = #/
      027510 020075 043
4091 027513 045 046123 020101 TMS1: .ASCII /%SLA #/
      027520 043
4092 027521 045 047502 020124 TMS2: .ASCII /%BOT #/
      027526 043
4093 027527 045 046524 021440 TMS3: .ASCII /%TM #/
4094 027534 044445 041104 021440 TMS4: .ASCII /%IDB #/
4095 027542 051445 053504 020116 TMS5: .ASCII /%SDWN #/
      027550 043
4096 027551 045 042520 020123 TMS6: .ASCII /%PES #/
      027556 043
4097 027557 045 051523 020103 TMS7: .ASCII /%SSC #/
      027564 043
4098 027565 045 051104 020131 TMS8: .ASCII /%DRY #/
      027572 043
4099 027573 045 050104 020122 TMS9: .ASCII /%DPR #/
      027600 043
4100 027601 045 052116 020114 TMS10: .ASCII /%NTL #/
      027606 043
4101 027607 045 047505 020124 TMS11: .ASCII /%EOT #/
      027614 043
4102 027615 045 051127 020114 TMS12: .ASCII /%WRL #/
      027622 043
4103 027623 045 047515 020114 TMS13: .ASCII /%MOL #/
      027630 043
4104 027631 045 044520 020120 TMS14: .ASCII /%PIP #/
      027636 043
4105 027637 045 051105 020122 TMS15: .ASCII /%ERR #/
      027644 043
4106 027645 045 052101 020101 TMS16: .ASCII /%ATA #/
      027652 043
4107 027653 045 046111 020106 TMS17: .ASCII /%ILF #/
      027660 043
4108 027661 045 046111 020122 TMS18: .ASCII /%ILR #/
      027666 043
4109 027667 045 046522 020122 TMS19: .ASCII /%RMR #/
      027674 043
4110 027675 045 050103 051101 TMS20: .ASCII /%CPAR #/
      027702 021440
4111 027704 043045 052115 021440 TMS21: .ASCII /%FMT #/
4112 027712 042045 040520 020122 TMS22: .ASCII /%DPAR #/
      027720 043
4113 027721 045 047111 020103 TMS23: .ASCII /%INC #/
      027726 043
4114 027727 045 050126 020105 TMS24: .ASCII /%VPE #/
      027734 043
4115 027735 045 042520 020106 TMS25: .ASCII /%PEF #/
      027742 043
4116 027743 045 051114 020103 TMS26: .ASCII /%LRC #/
      027750 043

```


4117	027751	045	051516	020107	TMS27:	.ASCII	/MSG #/
	027756	043					
4118	027757	045	041506	020105	TMS28:	.ASCII	/FCE #/
	027764	043					
4119	027765	045	051503	021440	TMS29:	.ASCII	/CS #/
4120	027772	044445	046524	021440	TMS30:	.ASCII	/ITM #/
4121	030000	047045	043105	021440	TMS31:	.ASCII	/NEF #/
4122	030006	042045	042524	021440	TMS32:	.ASCII	/DTE #/
4123	030014	047445	044520	021440	TMS33:	.ASCII	/OPI #/
4124	030022	053445	044522	042524	TMS33A:	.ASCII	/WRITE OPI #/
	030030	047440	044520	021440			
4125	030036	051045	040505	020104	TMS33B:	.ASCII	/READ OPI #/
	030044	050117	020111	043			
4126	030051	040	041517	052503	TMS33C:	.ASCII	/ OCCURED TO SOON##/
	030056	042522	020104	047524			
	030064	051440	047517	022516			
	030072	043					
4127	030073	040	041517	052503	TMS33D:	.ASCII	/ OCCURRED TO LATE##/
	030100	051122	042105	052040			
	030106	020117	040514	042524			
	030114	021445					
4128	030116	043040	044501	042514	TMS33E:	.ASCII	/ FAILED TO SET##/
	030124	020104	047524	051440			
	030132	052105	021445				
4129	030136	052445	051516	021440	TMS34:	.ASCII	/UNS #/
4130	030144	041445	051117	020122	TMS35:	.ASCII	/CORR #/
	030152	043					
4131	030153	045	051103	020103	TMS36:	.ASCII	/CRC #/
	030160	043					
4132	030161	045	040523	020103	TMS37:	.ASCII	/SAC #/
	030166	043					
4133	030167	045	041506	020123	TMS38:	.ASCII	/FCS #/
	030174	043					
4134	030175	045	041501	046103	TMS39:	.ASCII	/ACCL #/
	030202	021440					
4135							
4136							
4137							
4138							
4139	030204	000100					
4141	030204	177777					
(1)	030206	177777					
(1)	030210	177777					
(1)	030212	177777					
(1)	030214	177777					
(1)	030216	177777					
(1)	030220	177777					
(1)	030222	177777					
(1)	030224	177777					
(1)	030226	177777					
(1)	030230	177777					
(1)	030232	177777					
(1)	030234	177777					
(1)	030236	177777					
(1)	030240	177777					
(1)	030242	177777					

.EVEN
;WRITE BUFFER

WDATA:

(1)	030244	177777	-1
(1)	030246	177777	-1
(1)	030250	177777	-1
(1)	030252	177777	-1
(1)	030254	177777	-1
(1)	030256	177777	-1
(1)	030260	177777	-1
(1)	030262	177777	-1
(1)	030264	177777	-1
(1)	030266	177777	-1
(1)	030270	177777	-1
(1)	030272	177777	-1
(1)	030274	177777	-1
(1)	030276	177777	-1
(1)	030300	177777	-1
(1)	030302	177777	-1
(1)	030304	177777	-1
(1)	030306	177777	-1
(1)	030310	177777	-1
(1)	030312	177777	-1
(1)	030314	177777	-1
(1)	030316	177777	-1
(1)	030320	177777	-1
(1)	030322	177777	-1
(1)	030324	177777	-1
(1)	030326	177777	-1
(1)	030330	177777	-1
(1)	030332	177777	-1
(1)	030334	177777	-1
(1)	030336	177777	-1
(1)	030340	177777	-1
(1)	030342	177777	-1
(1)	030344	177777	-1
(1)	030346	177777	-1
(1)	030350	177777	-1
(1)	030352	177777	-1
(1)	030354	177777	-1
(1)	030356	177777	-1
(1)	030360	177777	-1
(1)	030362	177777	-1
(1)	030364	177777	-1
(1)	030366	177777	-1
(1)	030370	177777	-1
(1)	030372	177777	-1
(1)	030374	177777	-1
(1)	030376	177777	-1
(1)	030400	177777	-1
(1)	030402	177777	-1

4142

4143

4144

4145

4146 030404 000100

4148 030404 000000

(1) 030406 000000

(1) 030410 000000

;READ BUFFER

RDATA:

0

0

0

(1)	030412	000000	0
(1)	030414	000000	0
(1)	030416	000000	0
(1)	030420	000000	0
(1)	030422	000000	0
(1)	030424	000000	0
(1)	030426	000000	0
(1)	030430	000000	0
(1)	030432	000000	0
(1)	030434	000000	0
(1)	030436	000000	0
(1)	030440	000000	0
(1)	030442	000000	0
(1)	030444	000000	0
(1)	030446	000000	0
(1)	030450	000000	0
(1)	030452	000000	0
(1)	030454	000000	0
(1)	030456	000000	0
(1)	030460	000000	0
(1)	030462	000000	0
(1)	030464	000000	0
(1)	030466	000000	0
(1)	030470	000000	0
(1)	030472	000000	0
(1)	030474	000000	0
(1)	030476	000000	0
(1)	030500	000000	0
(1)	030502	000000	0
(1)	030504	000000	0
(1)	030506	000000	0
(1)	030510	000000	0
(1)	030512	000000	0
(1)	030514	000000	0
(1)	030516	000000	0
(1)	030520	000000	0
(1)	030522	000000	0
(1)	030524	000000	0
(1)	030526	000000	0
(1)	030530	000000	0
(1)	030532	000000	0
(1)	030534	000000	0
(1)	030536	000000	0
(1)	030540	000000	0
(1)	030542	000000	0
(1)	030544	000000	0
(1)	030546	000000	0
(1)	030550	000000	0
(1)	030552	000000	0
(1)	030554	000000	0
(1)	030556	000000	0
(1)	030560	000000	0
(1)	030562	000000	0
(1)	030564	000000	0
(1)	030566	000000	0
(1)	030570	000000	0

(1) 030572 000000
 (1) 030574 000000
 (1) 030576 000000
 (1) 030600 000000
 (1) 030602 000000

0
 0
 0
 0
 0

4149

4150

;WRAP AROUND MESSAGES*****

4151

4152

030604 042045 052101 020101

WMSG27: .ASCII /#DATA PAT:0/

030612 040520 035124 043

4153

030617 045 047505 020122

WMSG31: .ASCII /#EOR CLEAR DID NOT CLEAR GO#0/

030624 046103 040505 020122

030632 044504 020104 047516

030640 020124 046103 040505

030646 020122 047507 021445

4154

4155

PRE: .EVEN

4156

030654 000000

0

4159

030656 000000

0

(1)

030660 000000

0

(1)

030662 000000

0

(1)

030664 000000

0

(1)

030666 000000

0

(1)

030670 000000

0

(1)

030672 000000

0

(1)

030674 000000

0

(1)

030676 000000

0

(1)

030700 000000

0

(1)

030702 000000

0

(1)

030704 000000

0

(1)

030706 000000

0

(1)

030710 000000

0

(1)

030712 000000

0

(1)

030714 000000

0

(1)

030716 000000

0

(1)

030720 000000

0

(1)

030722 000000

0

(1)

030724 000000

0

(1)

030726 000000

0

(1)

030730 000000

0

(1)

030732 000000

0

(1)

030734 000000

0

(1)

030736 000000

0

(1)

030740 000000

0

(1)

030742 000000

0

(1)

030744 000000

0

(1)

030746 000000

0

(1)

030750 000000

0

(1)

030752 000000

0

(1)

030754 000000

0

(1)

030756 000000

0

(1)

030760 000000

0

(1)

030762 000000

0

(1)

030764 000000

0

(1)

030766 000000

0

(1)

030770 000000

0

(1)	030772	000000		0
(1)	030774	000000		0
4160	030776	000000	POST:	0
4163	031000	000000		0
(1)	031002	000000		0
(1)	031004	000000		0
(1)	031006	000000		0
(1)	031010	000000		0
(1)	031012	000000		0
(1)	031014	000000		0
(1)	031016	000000		0
(1)	031020	000000		0
(1)	031022	000000		0
(1)	031024	000000		0
(1)	031026	000000		0
(1)	031030	000000		0
(1)	031032	000000		0
(1)	031034	000000		0
(1)	031036	000000		0
(1)	031040	000000		0
(1)	031042	000000		0
(1)	031044	000000		0
(1)	031046	000000		0
(1)	031050	000000		0
(1)	031052	000000		0
(1)	031054	000000		0
(1)	031056	000000		0
(1)	031060	000000		0
(1)	031062	000000		0
(1)	031064	000000		0
(1)	031066	000000		0
(1)	031070	000000		0
(1)	031072	000000		0
(1)	031074	000000		0
(1)	031076	000000		0
(1)	031100	000000		0
(1)	031102	000000		0
(1)	031104	000000		0
(1)	031106	000000		0
(1)	031110	000000		0
(1)	031112	000000		0
(1)	031114	000000		0
(1)	031116	000000		0
4164	031120	000000	WBUFF:	0
4165		031532		. . .410
4166	031532	000000	RBUFF:	0
4167				
4168		000001		.END

G10

CZTEAF0 IM03 TE16/TU77 CTL I
CZTEAF.P11 06 APR 84 09:45

MACY11 30(1046) 06-APR-84 09:48 PAGE 89-1
CROSS REFERENCE TABLE USER SYMBOLS

SEQ 0123

ER	000524	1348#	1807	1839	2264	2293	2322	2347	2394	2407	2429	2465	2471	2477
		2496	2521	2547	2553	2582	2585	2605	2608	2730	2904	2909	2916	2954
		2975	2982	3012	3025	3056	3070	3102	3114	3148	3154	3216	3248	3382
		3410												
ERADD	000666	1406#	1786#	1816#	1819#	1822#	1848#	1853#	1932#	1966#	1970#	1996#	2021#	2049#
		2077#	2083#	2089#	2115#	2119#	2123#	2142#	2169#	2192#	2214#	2240#	2267#	2296#
		2325#	2350#	2401#	2433#	2473#	2499#	2512#	2569#	2594#	2658	2682	2693	2733#
		2760#	2769#	2792#	2819#	2840#	2880#	2901#	2906#	2911#	2977#	3014#	3023#	3058#
		3067#	3104#	3112#	3150#	3192#	3308#	3319	3500					
ERRF	000722	1420#												
EXFL	000712	1416#	1783#	1785#	1815#	1818#	1821#	1928#	1930#	2268#	2297#	2326#	2351#	2402#
		2434#	2474#	2498#	2523#	2549#	2734#	2761#	2777#	2784#	2802#	2810#	2824#	2831#
		2845#	2858#	2881#	2902#	2907#	2912#	2978#	3015#	3059#	3105#	3151#	3193#	3219#
		3251#	3307#	3325	3328	3362#								
FC	000516	1345#	1838#	1841	1850	2011#	2012	2066#	2106#	2291#	2319#	2370#	2425#	2454#
		2519#	2528#	2577#	2756#	2827#	2893#	2945#	2966#	3004#	3042#	3046	3088#	3092
		3133#	3136	3173#	3188	3208#	3240#	3376						
FUN	000746	1430#												
CTSWR	020620	1684	3682	3940#										
HDRFL	000614	1385#	1705#	3291	3314	3318#	3427	3469	3477#	3495	3499#			
HERE	002676	1734#												
ILFT	000544	1359#	2259											
INIT	017004	2943	2964	2998	3040	3086	3131	3606#						
INIT1	016726	1837	1942	1984	2010	2061	2072	2102	2111	2134	2163	2260	2287	2317
		2341	3583#											
INIT2	016736	2186	2208	2234	3585#									
INIT3	016754	2367	2423	2448	2494	2515	2525	2571	2595	2717	2754	2772	2794	2805
		2820	2826	2841	2848	2891	3171	3204	3595#					
INIT4	016770	2780	2868	3236	3600#									
INIT5	017010	3597	3602	3607#										
INMT	016650	2159	2182	2204	2230	3561#								
INST	017200	1755	1877	2162	2185	2207	2233	3640#						
ITAMT	000602	1377#	2366#	2416#	2570#	2616#	3553							
ITCNT	000676	1410#	3352	3516	3548#	3552#	3553							
ITER	016600	1828	1862	1974	2000	2026	2053	2094	2126	2145	2171	2194	2216	2242
		2278	2309	2332	2357	2415	2440	2481	2505	2558	2614	2740	2763	2786
		2812	2833	2860	2883	2925	2986	3030	3075	3120	3163	3195	3225	3257
		3546#												
ITRLP	000710	1415#	1703#	1704#	3556									
JUMPER	001016	1450#	1629	1634										
LTADD	000736	1426#	1701#	1702#	1703	1706	1722#	1723#						
LTGA	015246	3314#	3360											
LTGA1	015264	3315	3318#											
LTGA2	015316	3322	3325#											
LTGB	015344	3329	3331#											
LTGC	015350	3326	3332#											
LTGCO	015436	3336	3347#											
LTGD	015442	3334	3348#											
LTGER	015230	1788	1824	1934	2500	3310#								
LTGER0	015222	2269	2298	2327	2352	2403	2435	2475	2524	2550	2735	2762	2779	2785
		2804	2811	2825	2832	2847	2859	2882	2903	2908	2913	2979	3016	3060
		3106	3152	3220	3252	3309#								
LTGER1	016336	1851	1855	1968	1972	1998	2023	2051	2081	2087	2093	2170	2193	2215
		2241	3024	3068	3113	3194	3491#							
LTGER2	016330	2117	2121	2125	2144	3490#								
LTGER3	015210	2273	2302	2331	2356	2411	2439	2480	2557	2588	2611	2739	2919	2960

		2985	3029	3074	3119	3162	3224	3256	3307#
LTGX	015472	3313	3349	3354#					
LTGXA	015502	3355	3357#						
LTGXX	015520	3358	3361#						
LTG1A	016354	3495#	3526						
LTG1B	016372	3496	3499#						
LTG1C	016502	3513	3518#						
LTG1T	016452	3503	3512#						
LTG1X	016506	3494	3519#						
LTG1XX	016536	3524	3527#						
LTG1X1	016520	3521	3523#						
LT1	002736	1478	1479	1745#					
LT1A	003062	1753	1767#	1782	1787				
LT1B	003130	1771	1777#						
LT1C	003136	1775	1779#	1789					
LT1ER	003146	1776	1783#						
LT1ER1	003156	1778	1785#						
LT1ER2	003164	1784	1786#						
LT1G	003012	1756#	1780						
LT1G0	003002	1754#							
LT1X	003206	1751	1764	1773	1790#				
LT10	004622	1492	2032#						
LT10A	004640	2036#	2048	2050					
LT10A1	004636	2035#	2046						
LT10B	004674	2043#	2052						
LT10E1	004714	2042	2049#						
LT10IT	004624	1493	2033#						
LT10X	004736	2044	2053#						
LT11	004746	1494	2059#						
LT11B	004774	2065#	2082	2086					
LT11C	005034	2072#	2088	2092					
LT11E1	005054	2064	2077#						
LT11E2	005104	2071	2083#						
LT11E3	005134	2076	2089#						
LT11IT	004750	1495	2060#	2080					
LT11X	005162	2075	2094#						
LT12	005172	1496	2100#						
LT12B	005216	2105#	2118	2120					
LT12C	005254	2111#	2122	2124					
LT12E1	005272	2104	2115#						
LT12E2	005314	2110	2119#						
LT12E3	005336	2114	2123#						
LT12IT	005174	1497	2101#	2116					
LT12X	005356	2113	2126#						
LT13	005366	1498	2132#						
LT13A	005426	2139#	2140						
LT13E1	005432	2141#							
LT13IT	005376	1499	2134#	2143					
LT13X	005460	2135	2145#						
LT14	005470	1500	2155#						
LT14A	005512	2156	2160#						
LT14IT	005530	1501	2163#	2168					
LT14X	005570	2167	2171#						
LT14XX	005574	2158	2172#						
LT15	005600	1502	2178#						
LT15A	005622	2179	2183#						

CZTEAL0 TMO3 TE16 T077 CTL I
CZTEAL.P11 06 APR 84 09:45

MACY11 50(1046) 06 APR 84 09:48 PAGE 89 3
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0125

LT15IT	005640	1503	2186#	2191
LT15X	005700	2190	2194#	
LT15XX	005704	2181	2195#	
LT16	005710	1504	2200#	
LT16A	005732	2201	2205#	
LT16IT	005750	1505	2208#	2213
LT16X	006010	2212	2216#	
LT16XX	006014	2203	2217#	
LT17	006022	1506	2226#	
LT17A	006044	2227	2231#	
LT17IT	006062	1507	2234#	2239
LT17X	006122	2238	2242#	
LT17XX	006126	2229	2243#	
LT2	003212	1480	1481	1795#
LT2A	003252	1802#	1812	1827
LT2B	003272	1805	1807#	
LT2C	003306	1808	1810#	
LT2ERG	003366	1817	1820	1823#
LT2ER1	003316	1806	1815#	
LT2ER2	003334	1809	1818#	
LT2ER3	003352	1821#		
LT2IT	003214	1796#		
LT2LP	003402	1823	1826#	
LT2X	003406	1813	1828#	
LT20	006134	1508	2256#	
LT20A	006162	2257	2260#	2277
LT20B	006240	2266	2271#	
LT20C	006250	2270	2272	2274#
LT20IT	006150	1509	2258#	
LT20X	006262	2275	2278#	
LT21	006276	1510	2285#	
LT21A	006402	2295	2300#	
LT21B	006412	2299	2301	2304#
LT21IT	006312	1511	2286	2287#
LT21XA	006416	2305#	2306	
LT22	006442	1512	2315#	
LT22A	006542	2324	2329#	
LT22IT	006456	1513	2316	2317#
LT22X	006552	2328	2330	2332#
LT23	006566	1514	2339#	
LT23A	006666	2349	2354#	
LT23IT	006602	1515	2340	2341#
LT23X	006676	2353	2355	2357#
LT24	006716	1516	2364#	
LT24B	007044	2383#	2389	
LT24B0	007066	2386	2388#	
LT24C	007100	2392#	2393	
LT24D	007162	2395	2406#	
LT24IT	006732	1517	2365	2366#
LT24X	007206	2405	2410	2412#
LT25	007242	1518	2422#	
LT25A	007350	2431	2437#	
LT25IT	007250	1519	2423#	2432
LT25X	007360	2436	2438	2440#
LT26	007374	1520	2447#	
LT26IT	007402	1521	2448#	2472

LT26X	007574	2476	2479	2481#		
LT27	007610	1522	2488#			
LT27A	007644	2490	2494#	2504		
LT27B	007702	2497	2501#			
LT27IT	007634	1523	2492#			
LT27X	007712	2502	2505#			
LT27XX	007722	2489	2507#			
LT3	003416	1482	1833#			
LT3A	003436	1837#	1847	1849	1861	
LT3B	003456	1841#	1852	1854		
LT3C	003472	1844#	1856			
LT3ER1	003502	1840	1848#			
LT3ER2	003530	1843	1853#			
LT3IT	003420	1483	1834#			
LT3X	003552	1845	1857#			
LT3XX	003570	1858	1862#			
LT30	007726	1524	2512#			
LT30A	010030	2522	2525#			
LT30B	010072	2532#	2535			
LT30C	010132	2533	2542#			
LT30D	010150	2545#	2546			
LT30E	010200	2548	2552#			
LT30IT	007750	1525	2514	2515#		
LT30X	010224	2541	2551	2556	2558#	
LT31	010244	1526	2567#			
LT31A	010422	2593	2595#			
LT31IT	010252	1527	2568#			
LT31X	010526	2589	2591	2612	2614#	2662
LT32	011074	1528	2715#			
LT32IT	011110	1529	2716	2717#		
LT32X	011236	2736	2738	2740#		
LT32XX	011246	2728	2742#			
LT33	011252	1530	2752#			
LT33IT	011266	1531	2753	2754#		
LT33X	011344	2759	2763#			
LT34	011354	1532	2769#			
LT34A	011414	2775#				
LT34A1	011374	2772#	2778			
LT34B	011444	2776	2780#			
LT34C	011450	2781#	2783			
LT34IT	011370	1533	2771#			
LT34X	011500	2782	2786#			
LT34XX	011504	2787#				
LT35	011510	1534	2792#			
LT35A	011612	2801	2805#	2809		
LT35IT	011524	1535	2794#	2803		
LT35X	011652	2808	2812#			
LT36	011662	1536	2818#			
LT36IT	011676	1537	2820#	2823		
LT36X	011772	2829	2833#			
LT37	012002	1538	2839#			
LT37A	012056	2844	2848#	2857		
LT37B	012110	2853#	2856			
LT37IT	012016	1539	2841#	2846		
LT37X	012142	2854	2860#			
LT4	003600	1484	1485	1868#		

LT4A	003716	1875	1889#	1927	1933
LT4B	003760	1894	1898#		
LT4C	003766	1900#			
LT4D	004120	1896	1924#	1935	
LT4ERG	004146	1929	1931#		
LT4ER1	004130	1897	1928#		
LT4ER2	004140	1899	1930#		
LT4G	003646	1878#	1925		
LT4G0	003636	1876#			
LT4X	004176	1873	1886	1923	1936#
LT40	012152	1540	2866#		
LT40IT	012166	1541	2866	2868#	
LT40X	012262	2877	2883#		
LT40XX	012266	2884#			
LT41	012272	1542	2889#		
LT41IT	012306	1543	2889	2891#	
LT41X	012502	2914	2918	2920#	
LT42	012534	1544	2936#		
LT42A	012636	2950#	2953		
LT42B	012652	2951	2954#		
LT42B1	012700	2955	2960#		
LT42B2	012720	2959	2964#		
LT42C	012764	2971#	2974		
LT42D	013000	2972	2975#		
LT42E	013030	2976	2981#		
LT42IT	012572	1545	2942	2943#	
LT42X	013050	2980	2984	2986#	
LT43	013064	1546	2992#		
LT43C	013174	3008#	3011		
LT43D	013210	3009	3012#		
LT43E	013242	3013	3018#		
LT43F	013274	3022	3025#		
LT43IT	013122	1547	2996	2998#	
LT43X	013314	3017	3028	3030#	
LT43XX	013324	2993	3032#		
LT44	013330	1548	3036#		
LT44A	013414	3046#	3049		
LT44A1	013426	3047	3050#		
LT44B	013436	3052#	3055		
LT44C	013452	3053	3056#		
LT44D	013500	3057	3061#		
LT44E	013502	3062#			
LT44F	013542	3063	3070#		
LT44IT	013356	1549	3039	3040#	
LT44X	013562	3069	3073	3075#	
LT44XX	013572	3077#			
LT45	013576	1550	3082#		
LT45A	013704	3098#	3101		
LT45B	013720	3099	3102#		
LT45D	013750	3103	3107#		
LT45E	013662	3092#	3095		
LT45E1	013674	3093	3096#		
LT45F	014002	3111	3114#		
LT45IT	013624	1551	3085	3086#	
LT45X	014026	3118	3120#		
LT45XX	014036	3122#			

CZTEAEO TM03-TE16/TU77 CTL I
CZTEAE.P11 06-APR-84 09:45

MACY11 30(1046) 06-APR-84 09:48 PAGE 89-7
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0129

MSG10	021704	1853	3976#			
MSG11	021730	1848	3977#			
MSG12	021755	3339	3504	3978#		
MSG13	021764	3343	3508	3979#		
MSG14	021773	1966	3980#			
MSG15	022010	1970	3981#			
MSG16	022026	3350	3514	3982#		
MSG18	022036	1996	3983#			
MSG19	022055	2021	3192	3984#		
MSG2	021165	1756	3964#			
MSG2A	021226	1754	3965#			
MSG20	022074	2049	3985#			
MSG21	022126	2077	3986#			
MSG22	022164	2083	3987#			
MSG23	022205	2089	3988#			
MSG24	022240	2115	3989#			
MSG25	022266	2119	3990#			
MSG25A	022315	2123	3991#			
MSG26	022342	2142	3992#			
MSG27	022371	2169	2192	2214	2240	3993#
MSG3	021355	1786	3967#			
MSG30	022405	1918	3994#			
MSG31	022413	3455	3995#			
MSG32	022431	3451	3996#			
MSG33	022447	3458	3997#			
MSG34	022470	3998#				
MSG35	022505	3999#				
MSG36	022523	4000#				
MSG37	022541	4001#				
MSG4	021377	1816	1822	3968#		
MSG40	022557	3769	4002#			
MSG41	022563	1727	4003#			
MSG42	022602	3067	4004#			
MSG43	022617	3562	4005#			
MSG44	022723	1594	4007#			
MSG45	022745	1603	4008#			
MSG46	022767	3368	4009#			
MSG47	023073	3431	4011#			
MSG5	021415	1819	3969#			
MSG50	023125	2538	4012#			
MSG51	023144	3308	4013#			
MSG53	023173	3023	4014#			
MSG54	023205	3112	4015#			
MSG55	023216	2961	4016#			
MSG56	023313	1655	4017#			
MSG57	023340	1637	4018#			
MSG58	023517	1646	4021#			
MSG59	023542	3728	4022#			
MSG6	021435	3327	3970#			
MSG60	023544	3737	4023#			
MSG62	023546	1585	4024#			
MSG63	023614	3675	4025#			
MSG64	023642	1900	4026#			
MSG65	023660	1903	4027#			
MSG66	023665	1906	4028#			
MSG67	023672	1665	4029#			

