

TM03/TE16,TU77

TM03/TE16.TU77 CLTI
CZTEAC0

AH A792C MC

COPYRIGHT 77 79

FICHE 1 OF 1

NOV 1979

digital

MADE IN USA

.REM %

IDENTIFICATION

PRODUCT CODE: AC-A791C-MC
PRODUCT NAME: CZTEACO TM03-TE16/TU77 CONTROL LOGIC TEST PART 1
DATE CREATED: 14 MAY 79
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: J. G. ADAMS

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

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

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

COPYRIGHT (©) 1977,1979 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	3
2.	REQUIREMENTS	2
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 (RH)
- F. TE16 MAGTAPE TRANSPORT

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE.

4. STARTING PROCEDURE

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

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

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

4.1 AUTOMATIC MODE OPERATION

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

**EXCEPTION: IF THIS PROGRAM IS LOADED VIA TMDP CHAIN MODE THE
PROGRAM WILL NOT TEST TMO3 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 RESONSES ARE SHOWN IN PARENTHESES (), AND
MEMORY LOCATIONS CONTAINING THE DEFAULT ARE SHOWN IN
SQUARE BRACKETS [].
IN THIS EXAMPLE THE OPERATOR HAS CHOSEN DEFAULT RESPONSES.
TO INVOKE THE DEFAULT TYPE (CR).

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

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

TMO3-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:]
TMO3 DRIVE: <0> (CR) [DRVN:]
TE16 SLAVE: <0> (CR) [SLVN:]
STATIC TESTS ONLY: <0> (CR) [STATC:]
SLAVE TYPE (0=TE16,1=TU77): <0> (CR) [SLVTYP:]
IF THE SOFTWARE SWR IS INVOKED:
SWR = <000000> NEW = (CR) [SWREG:]

5. CONSOLE SWITCH SETTINGS

CONTROL:

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

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

SW15: 1=HALT ON ERROR
0=CONTINUE
SW14: 1=LOOP ON ERROR (SCOPE)
0=CONTINUE
SW13: 1=DO NOT PRINT ERRORS
0=PRINT ALL ERRORS
SW12: 1=DO CONTINUOUS CYCLE
0=HALT AT END OF PASS
SW11: 1=INHIBIT ITERATIONS
0=ITERATE EACH TEST ITS ASSIGNED AMOUNT
SW10: 1=HALT AT END OF CURRENT TEST
0=CONTINUE TO NEXT TEST
SW9: 1=DO MANUAL 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)
UNEXPTED 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	(MB12)
DEM-TRA HANDSHAKE	

LOGIC TEST #2: REGISTER ADDRESSING

PURPOSE: CHECK THE REGISTER SELECT LINES

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

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

CIRCUITS

PRINT REFERENCE

C-LINES	(MB1,2,3),(MB13),(MB14),(MB15)
RH REGISTER SELECT	(BCTA)
TMO3 REGISTER SELECT	(MB12)
MASSBUS REGISTER SELECT LINES	(MB1,2)
PARITY TREE	(MB14)
CPAR,ILR BITS	(MB111)

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	(MB13,4,5,8)(TCCM7)(MR)
ERROR BIT	(MB111)
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	(MB12)
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	(MB12)
TC FLIP-FLOPS, MULTIPLEXERS	(MR6)

LOGIC TEST #7: FRAME COUNT BIT TEST

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

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

LIKELY FAULT LOCATIONS: M8909

CIRCUITS

PRINT REFERENCE

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

LOGIC TEST #10: FUNCTION CODE BIT TEST

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

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

LIKELY FAULT LOCATION: M8909, M8905-YB

CIRCUITS

PRINT REFERENCE

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

LOGIC TEST #11: GO BIT SET, RESET

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

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

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

CIRCUIT -----	PRINT REFERENCE -----
FCS	MB18
SET ILF	MB17
SET NEF	MB17
GO BIT	MB15
GO BIT MULTIPLEXER	MR6
SET ILR	MB12

LOGIC TEST #12: DRIVE READY BIT

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

PRINT REF TCCM7

LOGIC TEST #13: INTERRUPT TEST

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

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

LIKELY FAULT LOCATION:

<u>CIRCUITS</u>	<u>PRINT REFERENCE</u>
-----------------	------------------------

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

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

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
MB13

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

SET ILF DECODE
ILF FLOP
ILF MULTIPLEXER

PRINT REFERENCE

MB15,MB17
MB111
MB110

LOGIC TEST #21: REGISTER MODIFICATION REFUSED

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

LIKELY FAULT LOCATION: M8909

CIRCUIT -----	PRINT REFERENCE -----
RMR DECODE	MB12
RMR FLOP	MB111
RMR MULTIPLEXER	MB110

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	MB14
CPAR FLOP	MB111
CPAR MULTIPLEXER	MB110

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	MB111
NEF MULTIPLEXER	MB110
SET NEF	MB17

LOGIC TEST #26: FRAME COUNT ERROR

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

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

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

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

LOGIC TEST #27: ILLEGAL REGISTER

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

LIKELY FAULT LOCATIONS: MASSBUSS, M8909

CIRCUITS

PRINT REFERENCE

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

LOGIC TEST #30: DRIVE TIMING ERROR

PROGRAMMED SEQUENCE:

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

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

CIRCUITS

PRINT REFERENCES

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

LOGIC TEST 31: OPERATION INCOMPLETE (OPI)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8933, M8909

CIRCUITS

PRINT REFERENCES

OPI TIMER
OPI FLOP
OPI TIMER CONTROL

TCCM5
MB111
MB17

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

MB111
MB17
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	MB15
PIP GENERATION	TCCM7
STATUS REGISTER	TCCM7

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905-YB, SLAVE BUS, 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
FCS MULTIPLEXER

MB18
TCCM7

LOGIC TEST 37: ACCELERATION (ACCL)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8933, M8931

CIRCUITS

PRINT REFERENCES

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

LOGIC TEST 40: PE TAPE MARK (TM)

PROGRAMMED SEQUENCE:

DRIVE IS INITIALIZED, WAMO IS SET, WRITE TAPE MARK IS SET. AFTER DELAY TAPE MARK BIT IS CHECKED. WAMO MULTIPLEXES THE OUTPUT OF THE WRITE DATA GENERATOR ONTO THE RDA LINES. THE DATA SYNC MODULES SYNC ON THE DATA AND SEND ENVELOPE INFORMATION TO THE TAPE MARK DETECTOR ON M8932.

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

CIRCUITS

PRINT REFERENCES

TAPE MARK DETECTOR	TCPE4, TCPE5
TAPE MARK MULTIPLEXER	TCCM7
ENVELOPE SIGNALS	DS 3, 5, 7
WRITE DATA BUFFER	TCCM2
RDA MULTIPLEXERS	TCCM6
WRITE TAPE MARK FUNCTION	MB15
WAMO SIGNAL	MR5

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

PROGRAMMED SEQUENCE:

SAME AS TEST 40 EXCEPT NRZ DENSITY IS SELECTED.

LIKELY FAULT LOCATIONS: M8933, M8934

CIRCUITS

PRINT REFERENCES

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

TCCM2
TCCM6
TCCM6
CNRZ4
CNRZ4

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

LOGIC TEST 42: CYCLIC REDUNDANCY ERROR

PROGRAMMED SEQUENCE:

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

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

CIRCUITS

PRINT REFERENCES

MM FUNCTION DECODE
CRC CHECK CIRCUIT

MR5
CNRZ3

LOGIC TEST 43: LRC

PROGRAMMED SEQUENCE:

A WRITE OPERATION IS PERFORMED WITH A MM FUNCTION (INC TMRL) WHICH ASSERTS WD(SB) 5L THROUGHOUT THE RECORD. ALL ONES DATA IS USED SO THAT THE FUNCTION ONLY INTERFERES WITH THE WRITING OF THE LRC CHARACTER WHEN NONE OF THE TM03 WRITE DATA LINES SHOULD BE ASSERTED.

** NOTE: THIS TEST IS NOT PERFORMED ON A TU77 SLAVE.

LIKELY FAULT LOCATIONS: M8505, M8933, M8910, M8934

CIRCUITS

PRINT REFERENCES

MM FUNCTION DECODE
WRITE LINE DRIVERS
WRITE HEAD DRIVERS
LRC CHECKING

MR5
TCCM2
LAW3, 4
CNRZ3

LOGIC TEST 44: PE CORRECTABLE DATA

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905-YB, M8901, M8932

CIRCUITS

PRINT REFERENCES

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

LOGIC TEST 45: PE INCORRECTABLE DATA

REPEAT OF TEST 44, EXCEPT THAT THE MAINT. MODE FUNCTION GROUNDS BITS STROBE FOR BITS 1, 2 AND THE WD LINE FOR BIT 5 IN HELD ASSERTED. INC. DATA AND PCF ERRORS ARE EXPECTED.

LIKELY FAULT LOCATIONS: M8932, M8901

CIRCUIT

PRINT REFERENCE

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

LOGIC TEST 46: PE FORMAT

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

LIKELY FAULT LOCATIONS: M8932, M8933, M8905-YB

CIRCUITS

PRINT REFERENCES

PEF.
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.

1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216

```
.LIST BIN,LOC,SEQ
.TITLE CZTEACO TMO3-TE16/TU77 CTL 1
:CONTROL LOGIC TEST PART I
:AC-A791B-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
: NON-STANDARD JUMPER MODE (SEE 4.2 IN DOC.)
.MCALL .SACT11,.$EOP,$CATCH,$SAVE,$RESTORE,$CHAIN,$SCHNMODE
.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=HALT AT END OF PASS
: 1=CONTINUOUS CYCLE
:SW11: 1=INHIBIT ITERATIONS
: 0=DO ITERATIONS
:SW10: 1=HALT AT END OF EACH TEST
: 0=CONTINUE
:SW9: 1=DO MANUAL INTERVENTION TESTS
: 0=INHIBIT MANUAL INTERVENTION
:SW0-5: SELECT TEST NUMBER :: 00=ALL TESTS
```



```
1263 ;REGISTER EQUIVS*****
1264
1265 000000 R0=%0
1266 000001 R1=%1
1267 000002 R2=%2
1268 000003 R3=%3
1269 000004 R4=%4
1270 000005 R5=%5
1271 000006 SP=%6
1272 000007 PC=%7
1273
1274
1275 ;ACT11 HOOK *****
1276 000764 $SVPC=, ;SAVE CURRENT LOCATION CTR
1277 000042 .=42
1278 000042 .WORD 0
1279 000046 .=46
1280 000046 .WORD $ENDAD ;SET LOCATION 46
1281 000052 .=52
1282 000052 .WORD 0 ;SET LOCATION 52 = 0
1283 000764 .= $SVPC ;RESTORE LOCATION CTR
1284
1285 ;TTY INTERRUPT VECTOR*****
1286
1287 000060 .=60
1288 000060 017254 .WORD TTINT ;TTY INTERRUPT HEADER ADDRESS
1289 000062 000340 .WORD 340 ;PRIORITY LEVEL 7
1290
1291 ;SOFTWARE SWITCH REGISTER*****
1292 ;USED IF HARDWARE SWR = 177777 OR NOT AVAILABLE
1293 000176 .=176
1294 000176 000000 SWREG: .WORD 0 ;SOFTWARE SWITCH REGISTER
1295
1296 ;START ADDRESS*****
1297 000200 .=200
1298 000200 000137 001332 JMP START ;PROGRAM START
1299
1300 ;RESTART ADDRESS*****
1301 000210 .=210
1302 000210 000137 002224 JMP ST2
1303
1304 ;TM03 INTERRUPT VECTOR*****
1305
1306 000224 .=224
1307 000224 017244 MTINT ;TAPE INTERRUPT HANDLER ADDRESS
1308 000226 000340 340
1309
```



```

1310
1311          000510          .-510
1312                                     :MASS BUS REGISTER EQUIVS*****
1313
1314 000510 172440          C1: 172440
1315 000512 172442          WC: 172442
1316 000514 172444          BA: 172444
1317 000516 172446          FC: 172446
1318 000520 172450          CS: 172450
1319 000522 172452          DS: 172452
1320 000524 172454          ER: 172454
1321 000526 172456          AS: 172456
1322 000530 172460          CC: 172460
1323 000532 172462          DB: 172462
1324 000534 172464          MR: 172464
1325 000536 172466          DT: 172466
1326 000540 172470          SN: 172470
1327 000542 172472          TC: 172472
1328
1329                                     :ILLEGAL FUNCTION CODES
1330
1331 000544 005405          ILFT: 5405
1332 000546 007415          7415
1333 000550 016423          16423
1334 000552 020437          20437
1335 000554 022443          22443
1336 000556 025447          25447
1337 000560 031455          31455
1338 000562 033465          33465
1339 000564 036473          36473
1340
1341                                     :CONSTANTS*****
1342
1343 000566 177776          PSW: 177776          :PROCESSOR STATUS
1344 000570 177570          SWR: 177570          :SWITCH REGISTER
1345 000572 177560          TKS: 177560          :TTY READER STATUS
1346 000574 177562          TKB: 177562          :TTY READ BUFFER
1347 000576 177564          TPS: 177564          :TTY PUNCH STATUS
1348 000600 177566          TPB: 177566          :TTY PUNCH BUFFER
1349 000602 000020          ITAMT: 20          :ITERATION AMOUNT
1350 000604 000224          VECT: 224          :INTERRUPT VECTOR(RH)
1351 000606 172440          REGS: 172440          :STARTING REGISTER ADDRESS
  
```

```
1352 ;FLAGS AND COUNTERS*****
1353
1354 000610 000000 TOB: 0
1355 000612 000000 TIB: 0
1356 000614 000000 HDRFL: 0
1357 000616 000000 EMADDR: 0
1358 000620 000000 DRVN: 0
1359 000622 000000 TR00: 0
1360 000624 000000 TR01: 0
1361 000626 000000 TR02: 0
1362 000630 000000 TR03: 0
1363 000632 000000 TR04: 0
1364 000634 000000 TR05: 0
1365 000636 000000 TR06: 0
1366 000640 000000 TR07: 0
1367 000642 000000 TR10: 0
1368 000644 000000 TR11: 0
1369 000646 000000 TR12: 0
1370 000650 000000 TR13: 0
1371 000652 000000 TR14: 0
1372 000654 000000 TR15: 0
1373 000656 000000 NRZOF: 0
1374 000660 000000 SLVN: 0
1375 000662 000000 PFLG: 0
1376 000664 000000 RTRN: 0
1377 000666 000000 ERADD: 0
1378 000670 000000 TEMP1: 0
1379 000672 000000 TEMP2: 0
1380 000674 000000 TEMP3: 0
1381 000676 000000 ITCNT: 0
1382 000700 000000 SAV1: 0
1383 000702 000000 SAV2: 0
1384 000704 000000 SAV3: 0
1385 000706 000000 SCOLP: 0
1386 000710 000000 ITRLP: 0
1387 000712 000000 EXFL: 0
1388 000714 000000 ATAF: 0
1389 000716 000000 SLAF: 0
1390 000720 000000 SSCF: 0
1391 000722 000000 ERRF: 0
1392 000724 000000 ASF: 0
1393 000726 000000 SCF: 0
1394 000730 000000 TREF: 0
1395 000732 000000 PEXFL: 0
1396 000734 000000 STFLG: 0
1397 000736 000000 LTADD: 0
1398 000740 000000 T24FL: 0
1399 000742 000000 ADDFL: 0
1400 000744 000000 WAM: 0
1401 000746 000000 FUN: 0
1402 000750 000000 DATC: 0
1403 000752 000000 WTAD: 0
1404 000754 000000 DATAD: 0
1405 000756 000000 RDAD: 0
1406 000760 000000 W2FLG: 0
1407 000762 000000 DERFL: 0
```

1408	000764	000000	PREFL:	0	
1409	000766	000000	SERFL:	0	
1410	000770	000000	CRCNT:	0	
1411	000772	000000	UDES:	0	
1412	000774	000000	WPGFL:	0	
1413	000776	000000	PATRN:	0	
1414	001000	000000	STATF:	0	
1415	001002	000000	RDRVF:	0	
1416	001004	000000	RCDP:	0	
1417	001006	000000	STATC:	0	
1418	001010	000000	SLVTYP:	.WORD 0	;++B INDICATES SLAVE TYPE (0/1 = TE16/TU77)
1419	001012	000000	SKAT:	0	
1420	001014	000000	PCNTR:	0	:PASS COUNTER
1421	001016	000003	JUMPER:	3	:+INDICATOR FOR NON-STANDARD CONFIG.
1422	001020	000000	NONSTD:	0	:+FLAG FOR NON-STANDARD CONFIG.
1423					
1424					:EXPT WRAP STATUS*****
1425					
1426	001022	000000	WCS1:	0	
1427	001024	000000	WCS2:	0	
1428	001026	000000	WDS:	0	
1429	001030	000000	WER:	0	
1430					
1431					:CORE DUMP PATTERNS*****
1432					
1433	001032	000005	WCDP2:	5	
1434	001034	000005		5	
1435	001036	000012		12	
1436	001040	000012		12	
1437	001042	000000		0	
1438	001044	000017	WCDPO:	17	
1439	001046	000017		17	
1440	001050	000017		17	
1441	001052	000017		17	
1442	001054	000000		0	

1443			
1444			;LOGIC TEST ENTRY TABLE*****
1445			
1446	001056	000000	TSTTBL: 0
1447	001060	000000	0
1448	001062	002734	LT1
1449	001064	002734	LT1
1450	001066	003210	LT2
1451	001070	003210	LT2
1452	001072	003414	LT3
1453	001074	003416	LT31T
1454	001076	003576	LT4
1455	001100	003576	LT4
1456	001102	004200	LT5
1457	001104	004206	LT51T
1458	001106	004370	LT6
1459	001110	004372	LT61T
1460	001112	004504	LT7
1461	001114	004506	LT71T
1462	001116	004620	LT10
1463	001120	004622	LT101T
1464	001122	004744	LT11
1465	001124	004746	LT111T
1466	001126	005170	LT12
1467	001130	005172	LT121T
1468	001132	005364	LT13
1469	001134	005374	LT131T
1470	001136	005466	LT14
1471	001140	005526	LT141T
1472	001142	005576	LT15
1473	001144	005636	LT151T
1474	001146	005706	LT16
1475	001150	005746	LT161T
1476	001152	006020	LT17
1477	001154	006060	LT171T
1478	001156	006132	LT20
1479	001160	006146	LT201T
1480	001162	006274	LT21
1481	001164	006310	LT211T
1482	001166	006440	LT22
1483	001170	006454	LT221T
1484	001172	006564	LT23
1485	001174	006600	LT231T
1486	001176	006714	LT24
1487	001200	006730	LT241T
1488	001202	007240	LT25
1489	001204	007246	LT251T
1490	001206	007372	LT26
1491	001210	007400	LT261T
1492	001212	007606	LT27
1493	001214	007632	LT271T
1494	001216	007724	LT30
1495	001220	007746	LT301T
1496	001222	010242	LT31
1497	001224	010250	LT311T
1498	001226	011072	LT32

1499	001230	011106	LT32IT
1500	001232	011250	LT33
1501	001234	011264	LT33IT
1502	001236	011352	LT34
1503	001240	011366	LT34IT
1504	001242	011506	LT35
1505	001244	011522	LT35IT
1506	001246	011660	LT36
1507	001250	011674	LT36IT
1508	001252	012000	LT37
1509	001254	012014	LT37IT
1510	001256	012150	LT40
1511	001260	012164	LT40IT
1512	001262	012270	LT41
1513	001264	012304	LT41IT
1514	001266	012532	LT42
1515	001270	012570	LT42IT
1516	001272	013062	LT43
1517	001274	013120	LT43IT
1518	001276	013326	LT44
1519	001300	013354	LT44IT
1520	001302	013574	LT45
1521	001304	013622	LT45IT
1522	001306	014040	LT46
1523	001310	014066	LT46IT
1524	001312	014302	LT47
1525	001314	014316	LT47IT
1526	001316	014472	LT50
1527	001320	014506	LT50IT
1528	001322	014646	LT51
1529	001324	014662	LT51IT
1530	001326	002620	
1531	001330	000051	

TADX: .WORD TEND
TLAST: .WORD 51

;CONTAIN: # OF TESTS

```

1532                                     .EVEN
1533                                     ;PROGRAM START AND HOUSEKEEPING*****
1534
1535                                     ;NOTE: PROGRAM STARTS HERE ON START AT 200
1536 001332 012706 000500 START: MOV #500,SP ;SET STACK POINTER
1537 001336 013746 000004 MOV @#4,-(SP) ;SAVE ERROR TRAP VECTOR
1538 001342 013746 000006 MOV @#6,-(SP) ;AND VECTOR +2
1539 001346 012737 001372 000004 MOV #1$,@#4 ;SET NEW VECTOR
1540 001354 005037 000006 CLR @#6 ;AND PSW
1541 001360 022777 177777 177202 CMP #-1,@SWR ;USE SOFTWARE SWITCH IF HARDWARE
1542 001366 001402 BEQ 2$ ;IS = 177777
1543 001370 000404 BR 3$ ;OTHERWISE USE HARDWARE SWR
1544 001372 022626 1$: CMP (SP)+,(SP)+ ;RESET STACK PTR
1545 001374 012737 000176 000570 2$: MOV #SWREG,SWR ;SET SOFTWARE SWITCH REGISTER
1546 001402 012637 000006 3$: MOV (SP)+,@#6 ;RESTORE ERROR TRAP VECTORS
1547 001406 012637 000004 MOV (SP)+,@#4
1548 001412 005037 001012 CLR SKAT ;CLEAR SKIP ADDRESS TEST FLAG
1549 001416 005027 CLR (PC)+ ;CLEAR CHAIN INDICATOR
1550 001420 000000 CHNFLG: .WORD 0 ;CHAIN MODE INDICATOR
1551 ;1/0 = CHAIN/NOT CHAIN MODE
1552 001422 005737 000042 TST @#42 ;BRANCH IF IN DUMP MODE
1553 001426 001407 BEQ 50$
1554 001430 012737 000176 000570 MOV #SWREG,SWR ;INVOKE SOFTWARE SWR
1555 001436 005237 001420 INC CHNFLG ;SET CHNFLG = CHAIN MODE
1556 001442 000137 002244 JMP TSCD ;GO TO CHAIN ADDRESS
1557 001446 50$:
1558 001446 000240 SCHN: NOP
1559 001450 122737 000006 000041 4$: CMPB #6,@#41 ;BRANCH IF NOT LOADED VIA TMDP
1560 001456 001005 BNE 5$
1561 001460 012704 023426 MOV #MSG62,R4 ;ADVISE USER TO REMOVE TMDP FROM
1562 001464 004737 017724 JSR PC,TTOUT ;UNIT UNDER TEST
1563 001470 000000 HALT
1564 001472 012704 020642 5$: MOV #MSG1,R4
1565 001476 004737 017724 JSR PC,TTOUT ;PRINT TITLE
1566 001502 005737 001420 TST CHNFLG ;SEE IF IN CHAIN MODE
1567 001506 001402 BEQ 6$ ;IF NOT: BR
1568 001510 000137 002244 JMP TSCD ;ELSE GO TO START OF TESTS
1569 001514 112737 000043 020642 6$: MOVB #'#,MSG1 ;DO NOT PRINT TITLE ON RESTART
1570 001522 012704 022603 MOV #MSG44,R4
1571 001526 004737 017724 JSR PC,TTOUT ;REQUEST REGISTER ADDRESS
1572 001532 013703 000606 MOV REGS,R3
1573 001536 004737 020052 JSR PC,OC1P ;PRINT CURRENT ADDRESS
1574 001542 012705 000606 MOV #REGS,R5 ;SET ADDRESS SAVE LOC
1575 001546 012701 000007 MOV #7,R1 ;SET SIZE OF RESPONSE
1576 001552 012702 176400 MOV #176400,R2 ;SET UPPER LIMIT
1577 001556 012703 172300 MOV #172300,R3 ;SET LOWER LIMIT
1578 001562 004737 017402 JSR PC,TTR ;GO GET RESPONSE
1579 001566 012704 022625 MOV #MSG45,R4
1580 001572 004737 017724 JSR PC,TTOUT ;REQUEST VECTOR
1581 001576 013703 000604 MOV VECT,R3
1582 001602 004737 020052 JSR PC,OC1P ;PRINT CURRENT VECTOR
1583 001606 012705 000604 MOV #VECT,R5 ;SET ADDRESS SAVE LOC
1584 001612 012701 000004 MOV #4,R1 ;SET SIZE OF RESPONSE
1585 001616 012702 000224 MOV #224,R2 ;SET UPPER LIMIT
1586 001622 012703 000150 MOV #150,R3 ;SET LOWER LIMIT
1587 001626 004737 017402 JSR PC,TTR ;GO GET RESPONSE

```

1588	001632	013700	000604	MOV	VECT,R0	:GET VECTOR
1589	001636	012720	017244	MOV	#MTINT,(R0)+	:LOAD INTERRUPT ADDRESS IN VECTOR
1590	001642	012710	000340	MOV	#340,(R0)	:LOAD PRIORITY
1591	001646	013700	000606	MOV	REGS,R0	:GET START OF REGS
1592	001652	012701	000016	MGV	#16,R1	:SET NUMBER OF REGS
1593	001656	012702	000510	MOV	#C1,R2	:GET START OF TABLE
1594	001662	010022		STO: MOV	R0,(R2)+	:BUILD TABLE
1595	001664	062700	000002	ADD	#2,R0	:BUMP ADDRESS
1596	001670	005301		DEC	R1	:SEE IF DONE
1597	001672	001373		BNE	STO	:IF NOT: BR
1598	001674	012702	000610	MGV	#TOB,R2	
1599	001700	012700	000077	MOV	#77,R0	
1600	001704	005022		ST1: CLR	(R2)+	:CLEAR FLAGS + COUNTERS
1601	001706	005300		DEC	R0	
1602	001710	001375		BNE	ST1	
1603	001712	012704	023236	MOV	#MS57A,R4	:+REQUEST IF JUMPER IS IN NON-STANDARD MODE
1604	001716	004737	017724	JSR	PC,TTOUT	:+
1605	001722	012705	001016	MOV	#JUMPER,R5	:+
1606	001726	012703	000000	MOV	#0,R3	:+LIMIT RESPONSE
1607	001732	012701	000002	MOV	#2,R1	:+SET CHAR. NUMBER TO 1
1608	001736	012702	000004	MOV	#4,R2	:+SET RANGE 0 - 4
1609	001742	004737	017402	JSR	PC,TTR	:+GET RESPONSE
1610	001746	022737	000002 001016	CMP	#2,JUMPER	:TEST FOR NON-STANDARD JUMPER.
1611	001754	001002		BNE	1\$	
1612	001756	004737	016666	JSR	PC,NOST	:GO TO MODIFY SCHEDLAR
1613	001762	012704	023220	1\$: MOV	#MSG57,R4	:REQUEST TM03 DRIVE #
1614	001766	004737	017724	JSR	PC,TTOUT	
1615	001772	013703	000620	MOV	DRVN,R3	:GET CURRENT DRIVE #
1616	001776	004737	020052	JSR	PC,OCTP	:PRINT IT
1617	002002	012705	000620	MOV	#DRVN,R5	:TTR ROUTINE RETURNS USER VALUE TO (R5)
1618	002006	012701	000002	MOV	#2,R1	:LIMIT RESPONSE
1619	002012	012702	000007	MOV	#7,R2	:LIMIT RANGE TO 0-7
1620	002016	012703	000000	MOV	#0,R3	
1621	002022	004737	017402	JSR	PC,TTR	:GET USER RESPONSE
1622	002026	012704	023377	MOV	#MSG58,R4	:REQUEST TE16 SLAVE #
1623	002032	004737	017724	JSR	PC,TTOUT	
1624	002036	013703	000660	MOV	SLVN,R3	:GET CURRENT SLAVE #
1625	002042	004737	020052	JSR	PC,OCTP	:AND PRINT IT
1626	002046	012705	000660	MOV	#SLVN,R5	:TTR ROUTINE RETURNS RESPONSE TO (R5)
1627	002052	012701	000002	MOV	#2,R1	:LIMIT RESONSE TO 1 CHARACTER
1628	002056	012702	000007	MOV	#7,R2	:BETWEEN 0 AND 7
1629	002062	012703	000000	MOV	#0,R3	
1630	002066	004737	017402	JSR	PC,TTR	:GET USER RESPONSE
1631	002072	012704	023173	MOV	#MSG56,R4	
1632	002076	004737	017724	JSR	PC,TTOUT	:REQUEST STATIC ONLY
1633	002102	013703	001006	MOV	STATC,R3	:GET CURRENT VALUE
1634	002106	004737	020052	JSR	PC,OCTP	:AND TYPE IT
1635	002112	012705	001006	MOV	#STATC,R5	:SET ADDRESS OF STATIC FLAG
1636	002116	012701	000002	MOV	#2,R1	:SET SIZE OF RESPONSE
1637	002122	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
1638	002126	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
1639	002132	004737	017402	JSR	PC,TTR	:GET RESPONSE
1640						
1641	002136	012704	023552	MOV	#MSG67,R4	:++B REQUEST SLAVE TYPE 'TE16 OR TU77'
1642	002142	004737	017724	JSR	PC,TTOUT	:++B
1643	002146	013703	001010	MOV	SLVTYP,R3	:++B GET CURRENT SLAVE TYPE

1644	002152	004737	020052		JSR	PC,OCTP	;++B TYPE CURRENT VALUE
1645	002156	012705	001010		MOV	#SLVTYP,R5	;++B GET ADDRESS OF SLVTYP FLAG
1646	002162	012701	000002		MOV	#2,R1	;++B SET SIZE OF RESPONSE
1647	002166	012702	000001		MOV	#1,R2	;++B SET UPPER LIMIT
1648	002172	012703	000000		MOV	#0,R3	;++B SET LOWER LIMIT
1649	002176	004737	017402		JSR	PC,TTR	;++B GET RESPONSE
1650	002202	005737	001010		TST	SLVTYP	:IS IT A TU77
1651	002206	001406			BEQ	ST2	:BRANCH IF NOT
1652	002210	012737	116741	006016	MOV	#116741,STWD16	:SET UP TEST WORD FOR TEST 16
1653	002216	012737	110741	006130	MOV	#110741,STWD17	:SET UP TEST WORD FOR TEST 17
1654							
1655							
1656					:START	210	
1657	002224	012706	000500		ST2: MOV	#500,SP	:SET STACK PTR
1658	002230	005037	001002		CLR	RDRVF	:CLEAR REVERSE FLAG
1659	002234	005037	001014		CLR	PCNTR	:CLEAR PASS COUNTER
1660	002240	004737	020500		JSR	PC,GTSWR	:GET SWITCHES


```

1661
1662
1663
1664 002244 052777 000100 176320 TSCD: BIS #100,@TKS ;SET KEYBOARD INTERRUPT ENABLE
1665 002252 005737 000042 TST @#42 ;ACT MODE ?
1666 002256 001407 BEQ 1$ ;BRANCH IF NOT
1667 002260 032737 000004 172466 BIT #4,@#172466
1668 002266 001403 BEQ 1$
1669 002270 012737 000001 001010 MOV #1,SLVTYP
1670 002276 005037 000774 1$: CLR WPGFL ;CLEAR WRAP PATRN FLAG
1671 002302 005037 000734 CLR STFLG ;CLEAR SINGLE TEST FLAG
1672 002306 017700 176256 MOV @SWR,RO
1673 002312 042700 177700 BIC #177700,RO ;BRANCH IF SINGLE
1674 002316 001122 BNE STSCD ;TEST SELECTED
1675 002320 005737 001420 TST CHNFLG ;:BRANCH IF NOT IN CHAIN MODE
1676 002324 001457 BEQ TSCDA
1677 002326 012737 177777 000620 MOV #-1,DRVN ;;INITIALIZE DRIVE #
1678 002334 012737 177777 000660 NXTDRV: MOV #-1,SLVN ;;INITIALIZE SLAVE #
1679 002342 012777 000040 176150 1$: MOV #40,@CS ;;INIT CONTROLLER
1680 002350 005237 000620 INC DRVN ;;STEP DRIVE #
1681 002354 022737 000010 000620 CMP #10,DRVN ;;EXIT IF ALL DRIVES TESTED
1682 002362 001521 BEQ $DONE ;;FOR AVAILABILITY
1683 002364 013777 000620 176126 MOV DRVN,@CS ;;LOAD DRIVE #
1684 002372 005777 176112 TST @C1 ;;ACCESS DRIVE
1685 002376 032777 010000 176114 BIT #10000,@CS ;;BRANCH IF DRIVE NON EXISTANT
1686 002404 001356 BNE 1$ ;;(NED = 1)
1687 002406 005237 000660 NXTSLV: INC SLVN ;;STEP SLAVE # AND BRANCH
1688 002412 001011 BNE 1$ ;;IF NOT SLAVE 0
1689 002414 005737 000620 TST DRVN ;;BRANCH IF NOT DRIVE # 0
1690 002420 001006 BNE 1$
1691 002422 122737 000006 000041 CMPB #6,@#41 ;;BRANCH IF NOT TMDP
1692 002430 001002 BNE 1$
1693 002432 005237 000660 INC SLVN ;;STEP TO SLAVE # 1
1694 002436 022737 000010 000660 1$: CMP #10,SLVN ;;BRANCH IF ALL SLAVES TESTED
1695 002444 001733 BEQ NXTDRV ;;FOR AVAILABILITY
1696 002446 013777 000660 176066 MOV SLVN,@TC ;;LOAD SLAVE UNIT #
1697 002454 032777 002000 176054 BIT #2000,@DT ;;BRANCH IF SLAVE NOT
1698 002462 001751 BEQ NXTSLV ;;PRESENT (SPR = 0)
1699 002464 012737 001056 000736 TSCDA: MOV #TSTTBL,LTADD
1700 002472 062737 000004 000736 TSCD0: ADD #4,LTADD
1701 002500 013737 000736 000710 TSCD1: MOV LTADD,ITRLP
1702 002506 062737 000002 000710 ADD #2,ITRLP ;SET ITERATION ADDRESS
1703 002514 005037 000614 CLR HDRFL ;CLEAR PRINT HEADER FLAG
1704 002520 017700 176212 MOV @LTADD,RO ;SET POINTER TO TEST
1705 002524 000110 JMP (RO) ;GO TO TEST
1706 002526 032777 002000 176034 TSCD2: BIT #2000,@SWR ;SEE IF HALT ON TEST
1707 002534 001403 BEQ TSCD3 ;IF NOT: BR
1708 002536 000000 HALT
1709 002540 005037 000774 CLR WPGFL ;CLEAR WRAP DATA GENERATOR FLAG
1710 002544 005737 000734 TSCD3: TST STFLG ;SE IF SINGLE TEST
1711 002550 001750 BEQ TSCD0 ;IF NOT: BR
1712 002552 017700 176012 MOV @SWR,RO
1713 002556 042700 177700 BIC #177700,RO ;BRANCH IF ALL TESTS DESIRED
1714 002562 001630 BEQ TSCD ;IF SO: BR
1715 002564 012737 000001 000734 STSCD: MOV #1,STFLG ;SET SINGLE TEST FLAG
1716 002572 023700 001330 CMP TLAST,RO ;SEE IF EXCEEDED TESTS
  
```

```

1717 002576 002410          BLT      TEND          ;IF SO: BR
1718 002600 006300          ASL      RO
1719 002602 006100          ROL      RO          ;SET TABLE MODIFIER
1720 002604 012737 001056 000736  MOV      #TSTTBL,LTADD
1721 002612 060037 000736  ADD      RO,LTADD    ;SET TEST POINTER
1722 002616 000730          BR       TSCD1
1723 002620 005737 001420  TEND:   TST      CHNFLG   ;BRANCH IF IN CHAIN MODE
1724 002624 001270          BNE     NXTSLV     ;STEP TO NEXT SLAVE
1725 002626 012704 022443  $DONE:  MOV      #MSG41,R4
1726 002632 004737 017724  JSR     PC,TTOUT   ;PRINT END OF PASS
1727 002636 013703 001014  MOV     PCNTR,R3
1728 002642 004737 020052  JSR     PC,OC1P    ;PRINT PASS NUMBER
1729 002646 005000          CLR     RO         ;DELAY WAITING FOR
1730 002650 005300          1$:     DEC     RO
1731 002652 001376          BNE     1$
1732 002654 013700 000042  MOV     @#42,RO    ;GET ACT11 RETURN ADDRESS
1733 002660 001405          BEQ     HERE      ;BRANCH IF NOT ACT11
1734 002662 000005          RESET
1735 002664 004710          $ENDAD: JSR     PC,(RO)
1736 002666 000240          NOP
1737 002670 000240          NOP
1738 002672 000240          NOP
1739 002674 000240          HERE:   NOP
1740 002676 005737 001420  TST     CHNFLG    ;BRANCH IF IN CHAIN MODE
1741 002702 001005          BNE     TENDX
1742 002704 032777 010000 175656  BIT     #10000,@SWR ;SEE IF HALT ON PASS
1743 002712 001401          BEQ     TENDX     ;IF NOT: BR
1744 002714 000000          HALT
1745 002716 012737 000001 001012  TENDX:  MOV     #1,SKAT   ;SET SKIP ADDRESS TEST FLAG
1746 002724 005237 001014  INC     PCNTR     ;BUMP PASS COUNTER
1747 002730 000137 002244  JMP     TSCD      ;RESTART
1748          ;LOGIC TEST 1: DRIVE ADDRESSING*****
1749
1750 002734 012737 023672 000616  LT1:    MOV     #MSLT1,EMADDR ;++B SET ERROR MSG HDR ADDRESS
1751 002742 013737 000620 000674  MOV     DRVN,TEMP3 ;GET DRIVE # IO BE TESTED
1752 002750 013701 000620  MOV     DRVN,R1
1753 002754 005737 001012  TST     SKAT      ;SEE IF SKIP ADDRESS TESTS
1754 002760 001403          BEQ     1$        ;IF NOT: BR
1755 002762 005737 000734  TST     STFLG     ;SEE IF SINGLE TEST
1756 002766 001506          BEQ     LT1X     ;IF NOT: BR
1757 002770 032777 001000 175572  1$:     BIT     #1000,@SWR ;BRANCH IF MAN INTERVENTION
1758 002776 001430          BEQ     LT1A     ;NOT SELECTED
1759 003000 012704 021106  LT1G0:  MOV     #MSG2A,R4
1760 003004 004737 017176  JSR     PC,INST   ;PRINT TEST INSTRUCTIONS
1761 003010 012704 021045  LT1G:   MOV     #MSG2,R4
1762 003014 004737 017724  JSR     PC,TTOUT  ;REQUEST DRIVE NUMBER
1763 003020 012705 000674  MOV     #TEMP3,R5 ;TTR ROUTINE RETURNS RESPONSE TO (R5)
1764 003024 012701 000002  MOV     #2,R1
1765 003030 012702 000007  MOV     #7,R2
1766 003034 012703 000000  MOV     #0,R3
1767 003040 004737 017402  JSR     PC,TTR    ;GET DRIVE NUMBER
1768 003044 005737 000670  TST     TEMP1     ;SEE IF ANOTHER DRIVE
1769 003050 001455          BEQ     LT1X     ;IF NOT: BR
1770 003052 005001          CLR     R1        ;SELECT DRIVE 0
1771 003054 012700 000010  MOV     #10,RO    ;SET NUMBER OF DRIVES
1772 003060 012777 000040 175432  LT1A:  MOV     #40,@CS   ;INIT
  
```



```
1797                                     ;LOGIC TEST 2: REGISTER ADDRESSING*****
1798
1799 003210 000240                               LT2:  NOP
1800 003212 012777 000040 175300  LT2IT:  MOV    #40,@CS      ;INIT
1801 003220 013777 000620 175272      MOV    DRVN,@CS      ;SELECT DRIVE
1802 003226 012737 023746 000616      MOV    #MSLT2,EMADDR ;SAVE LT2 HEADER ADDRESS
1803 003234 012705 000510              MOV    #C1,R5        ;SET ADDRESS OF FIRST REGISTER
1804 003240 012700 000016              MOV    #16,R0        ;SET NUMBER OF REGISTERS
1805 003244 012702 000622              MOV    #TR00,R2      ;SET START OF REGISTER BUFFER
1806 003250 011501                               LT2A:  MOV    (R5),R1
1807 003252 011112                               MOV    (R1),(R2)     ;READ REGISTER
1808 003254 032777 020000 175226      BIT    #20000,@C1    ;SEE IF ERROR
1809 003262 001402                               BEQ    LT2B          ;IF NOT: BR
1810 003264 004737 003314              JSR    PC,LT2ER1     ;ELSE GO TO ERROR 1
1811 003270 032777 000002 175226  LT2B:  BIT    #2,@ER    ;SEE IF ILR
1812 003276 001402                               BEQ    LT2C          ;IF NOT: BR
1813 003300 004737 003332              JSR    PC,LT2ER2     ;ELSE GO TO ERROR 2
1814 003304 022225                               LT2C:  CMP    (R2)+,(R5)+ ;BUMP ADDRESS
1815 003306 005300                               DEC    R0
1816 003310 001357                               BNE    LT2A          ;CONTINUE FOR ALL REGISTERS
1817 003312 000434                               BR     LT2X
1818
1819 003314 012737 000002 000712  LT2ER1: MOV    #2,EXFL    ;FLAG NOT EXPECTED
1820 003322 012737 021257 000666      MOV    #MSG4,ERADD   ;POINT TO CONTROLLER ERROR
1821 003330 000415                               BR     LT2ERG        ;GO TO ERROR
1822 003332 012737 000002 000712  LT2ER2: MOV    #2,EXFL    ;FLAG NOT EXPECTED
1823 003340 012737 021275 000666      MOV    #MSG5,ERADD   ;POINT TO DRIVE ERROR
1824 003346 000406                               BR     LT2ERG        ;GO TO ERROR
1825 003350 012737 000001 000712  LT2ER3: MOV    #1,EXFL    ;FLAG EXPECTED
1826 003356 012737 021257 000666      MOV    #MSG4,ERADD   ;POINT TO DRIVE
1827 003364 012737 003400 000706  LT2ERG: MOV    #LT2LP,SCOLP ;SET SCOPE ADDRESS
1828 003372 004737 015226              JSR    PC,LTGER      ;GO PRINT
1829 003376 000207                               RTS    PC            ;ELSE CONTINUE
1830 003400 005726                               LT2LP: TST   (SP)+    ;RESET STACK
1831 003402 000722                               BR     LT2A          ;LOOP
1832 003404 004737 016576  LT2X:  JSR    PC,ITER   ;GO SEE IF ITERATIONS
1833 003410 000137 002526              JMP    TSCD2         ;RETURN TO SCHED
```

```

1834                                     ;LOGIC TEST 3: CONTROL BUS*****
1835
1836 003414 000240                      LT3:  NOP
1837 003416 012737 024025 000616      LT3IT: MOV    #MSLT3,EMADDR ;SET TEST HEADER
1838 003424 012701 000001                MOV    #1,R1 ;PRESET PATTERN 1
1839 003430 012700 000020                MOV    #20,R0 ;SET PATTERN CHANGE NUMBER
1840 003434 004737 016724                LT3A: JSR    PC,INIT1 ;GO INIT
1841 003440 010177 175052                MOV    R1,@FC ;WRITE TO FC
1842 003444 032777 000010 175052        BIT    #10,@ER ;SEE IF CPAR (TM03)
1843 003452 001012 - - - - -             BNE   LT3ER1 ;IF SO: BR
1844 003454 017702 175036                LT3B: MOV    @FC,R2 ;READ FC
1845 003460 032777 020000 175022        BIT    #20000,@C1 ;SEE IF MCPE (RH)
1846 003466 001017 - - - - -             BNE   LT3ER2 ;IF SO: BR
1847 003470 005300 - - - - -             LT3C: DEC    R0 ;SEE IF DONE PATTERN CHANGES
1848 003472 001426 - - - - -             BEQ   LT3X ;IF SO: BR
1849 003474 006301 - - - - -             ASL   R1 ;CHANGE PATTERN
1850 003476 000756 - - - - -             BR    LT3A ;CONTINUE
1851 003500 012737 021610 000666      LT3ER1: MOV   #MSG11,ERADD ;SET ERROR CODE
1852 003506 012737 003434 000706      MOV   #LT3A,SCOLP ;SET SCOPE ADDRESS
1853 003514 017702 174776                MOV   @FC,R2 ;GET DATA
1854 003520 004737 016334                JSR   PC,LTGER1 ;GO DO ERROR
1855 003524 000753 - - - - -             BR    LT3B
1856 003526 012737 021564 000666      LT3ER2: MOV   #MSG10,ERADD ;SET ERROR CODE
1857 003534 012737 003454 000706      MOV   #LT3B,SCOLP ;SET SCOPE ADDRESS
1858 003542 004737 016334                JSR   PC,LTGER1 ;GO DO ERROR
1859 003546 000750 - - - - -             BR    LT3C
1860 003550 105701 - - - - -             LT3X: TSTB  R1 ;SEE IF DONE PATTERN 2
1861 003552 100405 - - - - -             BMI   LT3XX ;IF SO: BR
1862 003554 012701 000401                MOV   #401,R1 ;SET PATTERN 2
1863 003560 012700 000010                MOV   #10,R0 ;SET PATTERN CHANGE NUMBER
1864 003564 000723 - - - - -             BR    LT3A ;DO PATTERN 2
1865 003566 004737 016576                LT3XX: JSR  PC,ITER ;GO SEE IF ITERATIONS
1866 003572 000137 002526                JMP   TSCD2 ;RETURN TO SCHEDULAR

```

```

1867
1868
1869
1870 003576 013737 000660 000674 LT4: MOV SLVN,TEMP3
1871 003604 013701 000660 MOV SLVN,R1
1872 003610 005737 001012 TST SKAT ;SEE IF SKIP ADDRESS TESTS
1873 003614 001403 BEQ 1$ ;IF NOT: BR
1874 003616 005737 000734 TST STFLG ;SEE IF SINGLE TEST
1875 003622 001564 BEQ LT4X ;IF NOT: BR
1876 003624 032777 001000 174736 1$: BIT #1000,@SWR ;BRANCH IF MAN INTERVETION
1877 003632 001430 BEQ LT4A ;NOT SELECTED
1878 003634 012704 021413 LT4G0: MOV #MSG8A,R4
1879 003640 004737 017176 JSR PC,INST ;PRINT TEST INSTRUCTIONS
1880 003644 012704 021352 LT4G: MOV #MSG8,R4
1881 003650 004737 017724 JSR PC,TTOUT ;REQUEST SLAVE
1882 003654 012705 000674 MOV #TEMP3,R5
1883 003660 012701 000002 MOV #2,R1
1884 003664 012702 000007 MOV #7,R2
1885 003670 012703 000000 MOV #0,R3
1886 003674 004737 017402 JSR PC,TTR ;GET SLAVE NUMBER
1887 003700 005737 000670 TST TEMP1 ;SEE IF SLAVE
1888 003704 001533 BEQ LT4X ;IF NOT: BR
1889 003706 005001 CLR R1 ;SELECT SLAVE 0
1890 003710 012700 000010 MOV #10,R0 ;SET NUMBER OF SLAVES
1891 003714 012777 000040 174576 LT4A: MOV #40,@CS ;INIT
1892 003722 013777 000620 174570 MOV DRVN,@CS ;SELECT DRIVE
1893 003730 010177 174606 MOV R1,@TC ;SELECT SLAVE
1894 003734 017703 174576 MOV @DT,R3 ;GET DT
1895 003740 020137 000674 CMP R1,TEMP3 ;SEE IF SHOULD HAVE SPR
1896 003744 001404 BEQ LT4B ;IF SO: BR
1897 003746 032703 002000 BIT #2000,R3 ;SEE IF SPR
1898 003752 001461 BEQ LT4D ;IF NOT: BR
1899 003754 000464 BR LT4ER1 ;GO TO ERROR 1
1900 003756 032703 002000 LT4B: BIT #2000,R3 ;SEE IF NO SLAVE PRESENT
1901 003762 001465 BEQ LT4ER2 ;(SPR=0)
1902 003764 012704 023522 LT4C: MOV #MSG64,R4 ;TYPE 'SLAVE TYPE = '
1903 003770 004737 017724 JSR PC,TTOUT
1904 003774 012702 000001 MOV #1,R2 ;PRESET SLAVE TYPE = TU77
1905 004000 012704 023540 MOV #MSG65,R4 ;SET UP TO TYPE 'TU77'
1906 004004 022777 142054 174524 CMP #142054,@DT ;BRANCH IF TU77
1907 004012 001412 BEQ 1$
1908 004014 012704 023545 MOV #MSG66,R4 ;CHANGE SLAVE TYPE TO 'TE16'
1909 004020 005302 DEC R2 ;CHANGE SLAVE TYPE TO TE16
1910 004022 022777 142051 174506 CMP #142051,@DT ;BRANCH IF TE16
1911 004030 001403 BEQ 1$
1912 004032 005302 DEC R2 ;CHENGE SLAVE TYPE TO ILLEGAL
1913 004034 012704 023662 MOV #MSG69,R4
1914 004040 004737 017724 1$: JSR PC,TTOUT ;TYPE SLAVE TYPE (TU77,TE16, OR ILLEGAL)
1915 004044 020237 001010 CMP R2,SLVTYP ;BRANCH IF HARDWARE SLAVE TYPE IS THE
1916 004050 001406 BEQ 4$ ;SAME AS USER SPACIFIED SLAVE TYPE
1917 004052 012704 023610 MOV #MSG68,R4 ;++B TYPE 'INCORRECT SLAVE TYPE'
1918 004056 004737 017724 JSR PC,TTOUT ;++B
1919 004062 000137 002620 JMP TEND ;++B EXIT TEST
1920 004066 012704 022265 4$: MOV #MSG30,R4
1921 004072 004737 017724 JSR PC,TTOUT ;PRINT SERIAL NUMBER TAG
1922 004076 017703 174436 MOV @SN,R3
  
```

1923	004102	004737	020376			JSR	PC,SNPT	;PRINT SERIAL NUMBER
1924	004106	032777	001000	174454		BIT	#1000,@SWR	;BRANCH IF NOT MANUAL INTERVENTION
1925	004114	001427				BEQ	LT4X	
1926	004116	005300			LT4D:	DEC	R0	
1927	004120	001651				BEQ	LT4G	;IF DONE ALL: BR
1928	004122	005201				INC	R1	;BUMP SLAVE
1929	004124	000673				BR	LT4A	;CONTINUE
1930	004126	012737	000001	000712	LT4ER1:	MOV	#1,EXFL	;FLAG EXPT: NOT RECEIVED
1931	004134	000403				BR	LT4ERG	
1932	004136	012737	000002	000712	LT4ER2:	MOV	#2,EXFL	;FLAG RECVD: NOT EXPT
1933	004144	012737	024112	000616	LT4ERG:	MOV	#MSLT4,EMADDR	;SET LT4 HEADER
1934	004152	012737	021542	000666		MOV	#MSG9,ERADD	;SET ERROR CONDITION
1935	004160	012737	003714	000706		MOV	#LT4A,SCOLP	;SET SCOPE ADDRESS
1936	004166	004737	015226			JSR	PC,LTGER	;GO TO ERROR
1937	004172	000751				BR	LT4D	;IF NO SCOPE: BR
1938	004174	000137	002526		LT4X:	JMP	TSCD2	;RETURN TO SCHED
1939								

```

1940                                     ;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST*****
1941
1942 004200 012737 024174 000616 LT5:  MOV    #MSLT5,EMADDR  ;SET TEST HEADER
1943 004206 004737 016724          LT5IT: JSR    PC,INIT1    ;GO INIT
1944 004212 012700 000032          MOV    #32,R0        ;SET LOOP FOR BITS 4-0
1945 004216 005001                  CLR    R1            ;SET TEST WORD
1946 004220 010177 174310          LT5A:  MOV    R1,@MR    ;SEND TEST WORD TO MR
1947 004224 017702 174304          MOV    @MR,R2       ;READ MR
1948 004230 042702 177740          BIC    #177740,R2   ;MASK BITS 4-0
1949 004234 020102                  CMP    R1,R2        ;SEE IF EXPT = RECVD
1950 004236 001026                  BNE   LT5ER1
1951 004240 005300          LT5B:  DEC    R0
1952 004242 001402          BEQ    LT5C          ;IF DONE LOOP: BR
1953 004244 005201          INC    R1            ;BUMP TEST WORD
1954 004246 000764          BR     LT5A          ;CONTINUE LOOP
1955 004250 012701 000015          LT5C:  MOV    #15,R1    ;SET TEST WORD + WAM 3
1956 004254 012700 001000          MOV    #1000,R0     ;SET LOOP FOR BITS 15-7
1957 004260 010177 174250          LT5D:  MOV    R1,@MR    ;LOAD MR
1958 004264 017702 174244          MOV    @MR,R2       ;READ MR
1959 004270 042702 000140          BIC    #140,R2      ;MASK OUT BITS 5,6
1960 004274 020102          CMP    R1,R2        ;SEE IF EXPT = RECVD
1961 004276 001401          BEQ    LT5E          ;IF SO: BR
1962 004300 000416          BR     LT5ER2       ;ELSE GO TO ERR 2
1963 004302 005300          LT5E:  DEC    R0
1964 004304 001425          BEQ    LT5X          ;IF DONE LOOP: BR
1965 004306 062701 000200          ADD    #200,R1      ;BUMP TEST WORD
1966 004312 000762          BR     LT5D          ;CONTINUE LOOP
1967 004314 012737 021653 000666 LT5ER1: MOV    #MSG14,ERADD ;SET ERROR CODE
1968 004322 012737 004220 000706 MOV    #LT5A,SCOLP  ;SET SCOPE ADDRESS
1969 004330 004737 016334          JSR    PC,LTGER1    ;GO TO ERROR
1970 004334 000741          BR     LT5B          ;CONTINUE
1971 004336 012737 021670 000666 LT5ER2: MOV    #MSG15,ERADD ;SET ERROR CODE
1972 004344 012737 004260 000706 MOV    #LT5D,SCOLP  ;SET SCOPE ADDRESS
1973 004352 004737 016334          JSR    PC,LTGER1    ;GO TO ERROR
1974 004356 000751          BR     LT5E          ;CONTINUE
1975 004360 004737 016576          LT5X:  JSR    PC,ITER    ;GO SEE IF ITERATIONS
1976 004364 000137 002526          JMP    TSCD2        ;RETURN TO SCHED
1977

```



```

1978                                     ;LOGIC TEST 6: TC REGISTER BIT TEST*****
1979
1980 004370 000240                       LT6:  NOP
1981 004372 012737 024243 000616 LT6IT: MOV  #MSLT6,EMADDR ;POINT TO LT6 HEADER
1982 004400 012700 000003                MOV  #3,R0 ;SET NUMBER OF TESTS
1983 004404 005001                       LT6A1: CLR  R1
1984 004406 004737 016724               LT6A:  JSR  PC,INIT1 ;GO INIT
1985 004412 010177 174124               LT6B:  MOV  R1,@TC ;WRITE TC
1986 004416 017702 174120                MOV  @TC,R2 ;READ TC
1987 004422 042702 160000                BIC  #160000,R2 ;MASK OUT SAC
1988 004426 020102                       CMP  R1,R2 ;SEE IF EXPT = RECVD
1989 004430 001010                       BNE  LT6ER1 ;IF NOT: BR
1990 004432 005300                       LT6D:  DEC  R0
1991 004434 001417                       BEQ  LT6X ;IF DONE ALL: BR
1992 004436 022700 000001               CMP  #1,R0 ;SEE IF RESET TEST
1993 004442 001760                       BEQ  LT6A1 ;IF SO: BR
1994 004444 012701 017777               MOV  #17777,R1 ;SET TEST WORD
1995 004450 000756                       BR   LT6A ;DO SET TEST
1996 004452 012737 021716 000666 LT6ER1: MOV  #MSG18,ERADD ;SET ERROR CODE
1997 004460 012737 004412 000706      MOV  #LT6B,SCOLP ;SET SCOPE ADDRESS
1998 004466 004737 016334                JSR  PC,LTGER1 ;GO TO ERROR
1999 004472 000757                       BR   LT6D ;CONTINUE
2000 004474 004737 016576               LT6X:  JSR  PC,ITER ;GO SEE IF ITERATIONS
2001 004500 000137 002526                JMP  TSCD2 ;RETURN TO SCHED
2002

```

```
2003                                     ;LOGIC TEST 7: FRAME COUNT BIT TEST*****
2004
2005 004504 000240                      LT7:  NOP
2006 004506 012700 000003                LT7IT: MOV    #3,R0          ;SET TEST NUMBER
2007 004512 012737 024312 000616        LT7C:  MOV    #MSLT7,EMADDR ;SET TEST HEADER
2008 004520 005001                      CLR    R1                ;SET TEST WORD
2009 004522 004737 016724                LT7A:  JSR    PC,INIT1     ;GO INIT
2010 004526 010177 173764                MOV    R1,@FC           ;CLEAR FRAME COUNT
2011 004532 017702 173760                MOV    @FC,R2          ;READ FC
2012 004536 020102                      CMP    R1,R2           ;SEE IF EXPT = RECVD
2013 004540 001010                      BNE    LT7ER1
2014 004542 005300                      LT7B:  DEC    R0          ;SEE IF DONE ALL
2015 004544 001417                      BEQ    LT7X            ;IF SO: BR
2016 004546 022700 000001                CMP    #1,R0           ;SEE IF RESET TEST
2017 004552 001757                      BEQ    LT7C            ;IF SO: BR
2018 004554 012701 177777                MOV    #-1,R1          ;SET TEST WORD TO -1
2019 004560 000760                      BR     LT7A            ;CONTINUE
2020 004562 012737 021735 000666        LT7ER1: MOV   #MSG19,ERADD ;SET ERROR CODE
2021 004570 012737 004522 000706        MOV   #LT7A,SCOLP     ;SET SCOPE ADDRESS
2022 004576 004737 016334                JSR   PC,LTGER1       ;GO PRINT ERROR
2023 004602 000757                      BR    LT7B            ;ELSE CONTINUE
2024 004604 012700 000003                LT7X:  MOV   #3,R0     ;RESET TEST AMT
2025 004610 004737 016576                JSR   PC,ITER         ;GO SEE IF ITERATIONS
2026 004614 000137 002526                JMP   TSCD2          ;RETURN TO SCHED
2027
```

```

2028                                     ;LOGIC TEST 10: FUNCTION CODE BIT TEST*****
2029
2030 004620 000240                      LT10:  NOP
2031 004622 012737 024361 000616      LT10IT: MOV #MSLT10,EMADDR ;SET TEST HEADER
2032 004630 012700 000003              MOV #3,R0 ;SET NUMBER OF TESTS
2033 004634 005001                      LT10A1: CLR R1 ;SET TEST WORRD
2034 004636 012777 000040 173654      LT10A:  MOV #40,@CS ;INIT
2035 004644 013777 000620 173646      MOV DRVN,@CS ;SELECT DRIVE
2036 004652 010177 173632              MOV R1,@C1 ;WRITE C1
2037 004656 017702 173626              MOV @C1,R2 ;READ C1
2038 004662 042702 177701              BIC #177701,R2 ;MASK FUNCTION CODE
2039 004666 020102                      CMP R1,R2 ;SEE IF EXPT = RECVD
2040 004670 001010                      BNE LT10E1
2041 004672 005300                      LT10B:  DEC R0
2042 004674 001417                      BEQ LT10X ;IF DONE ALL: BR
2043 004676 022700 000001              CMP #1,R0 ;SEE IF RESET TEST
2044 004702 001754                      BEQ LT10A1 ;IF SO: BR
2045 004704 012701 000076              MOV #76,R1 ;SET TEST WORD
2046 004710 000752                      BR LT10A ;DO SET TEST
2047 004712 012737 021754 000666      LT10E1: MOV #MSG20,ERADD ;SET ERROR CODE
2048 004720 012737 004636 000706      MOV #LT10A,SCOLP ;SET SCOPE ADDRESS
2049 004726 004737 016334              JSR PC,LTGER1 ;GO PRINT ERROR
2050 004732 000757                      BR LT10B ;ELSE CONTINUE
2051 004734 004737 016576              LT10X:  JSR PC,ITER ;GO SEE IF ITERATIONS
2052 004740 000137 002526              JMP TSCD2 ;RETURN TO SCHED
  
```

```
2053
2054
2055
2056 004744 000240
2057 004746 012737 024437 000616
2058 004754 004737 016724
2059 004760 017702 173524
2060 004764 032702 000001
2061 004770 001030
2062 004772 012777 000015 173534
2063 005000 005077 173512
2064 005004 052777 001700 173530
2065 005012 012777 000071 173470
2066 005020 017702 173464
2067 005024 032702 000001
2068 005030 001424
2069 005032 004737 016724
2070 005036 017702 173446
2071 005042 032702 000001
2072 005046 001444
2073 005050 000430
2074 005052 012737 022006 000666
2075 005060 012702 000001
2076 005064 005001
2077 005066 012737 004746 000706
2078 005074 004737 016334
2079 005100 000734
2080 005102 012737 022044 000666
2081 005110 005002
2082 005112 012701 000001
2083 005116 012737 004772 000706
2084 005124 004737 016334
2085 005130 000740
2086 005132 012737 022065 000666
2087 005140 005001
2088 005142 012702 000001
2089 005146 012737 005032 000706
2090 005154 004737 016334
2091 005160 004737 016576
2092 005164 000137 002526

;LOGIC TEST 11: GO BIT SET RESET*****
LT11: NOP
LT11IT: MOV #MSLT11,EMADDR ;SET TEST HEADER
JSR PC,INIT1 ;GO INIT
MOV @C1,R2 ;READ C1
BIT #1,R2 ;SEE IF GO=0
BNE LT11E1
LT11B: MOV #15,@MR ;SELECT WAM 3
CLR @FC ;ASSURE FCS = 1
BIS #1700,@TC ;ASSURE FMT OK
MOV #71,@C1 ;SET READ+GO
MOV @C1,R2 ;READ C1
BIT #1,R2 ;SEE IF GO =1
BEQ LT11E2
LT11C: JSR PC,INIT1 ;GO INIT
MOV @C1,R2 ;READ C1
BIT #1,R2 ;SEE IF GO=0
BEQ LT11X ;IF SO:BR
BR LT11E3 ;ELSE GO TO ERROR 3
LT11E1: MOV #MSG21,ERADD ;SET ERROR CODE
MOV #1,R2 ;SET REVD
CLR R1 ;SET EXPT
MOV #LT11IT,SCOLP ;SET SCOPE ADDRESS
JSR PC,LTGER1 ;GO PRINT ERROR
BR LT11B ;ELSE CONTINUE
LT11E2: MOV #MSG22,ERADD ;SET ERROR CODE
CLR R2 ;SET RCVD
MOV #1,R1 ;SET EXPT
MOV #LT11B,SCOLP ;SET SCOPE ADDRESS
JSR PC,LTGER1 ;GO PRINT ERROR
BR LT11C ;ELSE CONTINUE
LT11E3: MOV #MSG23,ERADD ;SET ERROR CODE
CLR R1 ;SET EXPT
MOV #1,R2 ;SET RCVD
MOV #LT11C,SCOLP ;SET SCOPE ADDRESS
JSR PC,LTGER1 ;GO PRINT ERROR
LT11X: JSR PC,ITER ;GO SEE IF ITERATIONS
JMP TSCD2 ;RETURN TO SCHED
```

```

2093
2094
2095
2096 005170 000240          LT12:  NOP
2097 005172 012737 024504 000616  LT12IT: MOV    #MSLT12,EMADDR ;SET TEST HEADER
2098 005200 004737 016724          JSR    PC,INIT1    ;GO INIT
2099 005204 032777 000200 173310  BIT    #200,ADS    ;SEE IF DRY=1
2100 005212 001426          BEQ    LT12E1
2101 005214 012777 000015 173312  LT12B: MOV    #15,@MR    ;SET WAM3
2102 005222 005077 173270          CLR    @FC        ;ASSURE FCS = 1
2103 005226 052777 001700 173306  BIS    #1700,@TC  ;ASSURE FMT OK
2104 005234 012777 000071 173246  MOV    #71,@C1    ;SET READ+GO
2105 005242 032777 000200 173252  BIT    #200,ADS    ;SEE IF DRY=0
2106 005250 001020          BNE    LT12E2
2107 005252 004737 016724          LT12C: JSR    PC,INIT1    ;GO INIT
2108 005256 032777 000200 173236  BIT    #200,ADS    ;SEE IF DRY=1
2109 005264 001033          BNE    LT12X
2110 005266 000422          BR     LT12E3      ;ELSE GO TO ERROR 3
2111 005270 012737 022120 000666  LT12E1: MOV    #MSG24,ERADD ;SET ERROR CODE
2112 005276 012737 005172 000706  MOV    #LT12IT,SCOLP ;SET SCOPE ADDRESS
2113 005304 004737 016326          JSR    PC,LTGER2  ;GO TO ERROR
2114 005310 000741          BR     LT12B      ;CONTINUE
2115 005312 012737 022146 000666  LT12E2: MOV    #MSG25,ERADD ;SET ERROR CODE
2116 005320 012737 005214 000706  MOV    #LT12B,SCOLP ;SET LOOP ADDRESS
2117 005326 004737 016326          JSR    PC,LTGER2  ;GO PRINT ERROR
2118 005332 000747          BR     LT12C      ;CONTINUE
2119 005334 012737 022175 000666  LT12E3: MOV    #MSG25A,ERADD ;SET ERROR CODE
2120 005342 012737 005252 000706  MOV    #LT12C,SCOLP ;SET ERROR LOOP
2121 005350 004737 016326          JSR    PC,LTGER2  ;GET PRINT ERROR
2122 005354 004737 016576          LT12X: JSR    PC,ITER  ;GO TO ITERATION SUBROUTINE
2123 005360 000137 002526          JMP    TSCD2      ;RETURN TO SCHED
  
```

```
2124
2125 ;LOGIC TEST 13: INTERRUPT TEST*****
2126
2127 005364 005000 LT13: CLR R0
2128 005366 012737 024555 000616 MOV #MSLT13,EMADDR ;SET TEST HEADER
2129 005374 004737 016724 LT13IT: JSR PC,INIT1 ;GO INIT,SELECT DRIVE, SELECT ABOVE
2130 005400 012737 005456 000664 MOV #LT13X,RTRN ;SET RETURN ADDRESS
2131 005406 005077 173076 CLR @C1 ;CLEAR CS1
2132 005412 005077 173150 CLR @PSW ;SET PRIORITY
2133 005416 052777 000100 173064 BIS #100,@C1 ;BIT SET IE
2134 005424 005300 LT13A: DEC R0
2135 005426 001376 BNE LT13A ;AWAIT INTERRUPT
2136 005430 012777 000340 173130 LT13E1: MOV #340,@PSW ;RESET PRIORITY
2137 005436 012737 022222 000666 MOV #MSG26,ERADD ;SET ERROR CODE
2138 005444 012737 005374 000706 MOV #LT13IT,SCOLP ;SET LOOP ADDRESS
2139 005452 004737 016326 JSR PC,LTGER2 ;GO PRINT ERROR
2140 005456 004737 016576 LT13X: JSR PC,ITER ;GO TO ITERATION SUBROUTINE
2141 005462 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167

: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*****

005466	032777	001000	173074	LT14:	BIT	#1000,@SWR	:SEE IF INHIB MAN TST
005474	001005				BNE	LT14A	:IF NOT: BR
005476	005737	000734			TST	STFLG	:SEE IF SINGLE TEST
005502	001433				BEQ	LT14XX	:IF NOT: BR
005504	000137	016646			JMP	INMT	:ELSE GO PRINT INHIB MSG
005510	012737	024622	000616	LT14A:	MOV	#MSLT14,EMADDR	:SET TEST HEADER
005516	012704	027027			MOV	#MMSG1,R4	:SET INSTRUCTION ONE
005522	004737	017176			JSR	PC,INST	:GO DO INSTRUCTION
005526	004737	016724		LT14IT:	JSR	PC,INIT1	:INIT, SELECT DRIVE + SLAVE
005532	012701	014602			MOV	#14602,R1	:SET TEST WORD
005536	017702	172760			MOV	@DS,R2	:ASSURE MOL,WRL,DPR,DRY,BOT
005542	020102				CMP	R1,R2	
005544	001410				BEQ	LT14X	:IF SO: BR
005546	012737	005526	000706		MOV	#LT14IT,SCOLP	:SET LOOP ADDRESS
005554	012737	022251	000666		MOV	#MSG27,ERADD	:SET ERROR CODE
005562	004737	016334			JSR	PC,LTGER1	:GO PRINT ERROR
005566	004737	016576		LT14X:	JSR	PC,ITER	:GO SEE IF ITERATION
005572	000137	002526		LT14XX:	JMP	TSCD2	:RETURN TO SCHED

```
2168
2169 ;LOGIC TEST 15: STATUS AT BOT, OFFLINE, LOADED, NO WRITE RING*****
2170
2171 005576 032777 001000 172764 LT15: BIT #1000,@SWR ;SEE IF INHIB MAN TST
2172 005604 001005 BNE LT15A ;IF NOT: BR
2173 005606 005737 000734 TST STFLG ;SEE IF SINGLE TEST
2174 005612 001433 BEQ LT15XX ;IF NOT: BR
2175 005614 000137 016646 JMP INMT ;ELSE GO PRINT INHIB MSG
2176 005620 012737 024731 000616 LT15A: MOV #MSLT15,EMADDR ;SET TEST HEADER
2177 005626 012704 027125 MOV #MMSG2,R4
2178 005632 004737 017176 JSR PC,INST ;PRINT INSTRUCTION
2179 005636 004737 016734 LT15IT: JSR PC,INIT2 ;GO INIT, SELECT DRIVE, SLAV
2180 005642 012701 100700 MOV #100700,R1 ;SET TEST WORD
2181 005646 017702 172650 MOV @DS,R2 ;READ STATUS
2182 005652 020102 CMP R1,R2 ;SEE OF EXPT=RCVD
2183 005654 001410 BEQ LT15X
2184 005656 012737 005636 000706 MOV #LT15IT,SCOLP ;SET LOOP ADDRESS
2185 005664 012737 022251 000666 MOV #MSG27,ERADD ;SET ERROR CODE
2186 005672 004737 016334 JSR PC,LTGER1 ;GO PRINT ERROR
2187 005676 004737 016576 LT15X: JSR PC,ITER ;GO SEE IF ITERATIONS
2188 005702 000137 002526 LT15XX: JMP TSCD2 ;RETURN TO SCHED
```



```

2189
2190 ;LOGIC TEST 16: STATUS AT EOT, ON-LINE, NO WRITE RING*****
2191
2192 005706 032777 001000 172654 LT16: BIT #1000,@SWR ;SEE IF INHIB MAN TST
2193 C05714 001005 BNE LT16A ;IF NOT: BR
2194 005716 005737 000734 TST STFLG ;SEE IF SINGLE TEST
2195 005722 001433 BEQ LT16XX ;IF NOT: BR
2196 005724 000137 016646 JMP INMT ;ELSE GO PRINT INHIB MSG
2197 005730 012737 025021 000616 LT16A: MOV #MSLT16,EMADDR ;SET TEST HEADER
2198 005736 012704 027146 MOV #MMSG3,R4
2199 005742 004737 017176 JSR PC,INST ;GO PRINT INSTRUCTION
2200 005746 004737 016734 LT16IT: JSR PC,INIT2 ;SELECT DRIVE,SLAVE
2201 005752 013701 006016 MOV STWD16,R1 ;SET TEST WORD (ATA!MOL!WRL!EOT!DPR!DRY!SSC!SLA)
2202 005756 017702 172540 MOV @DS,R2 ;READ STATUS
2203 005762 020102 CMP R1,R2 ;SEE IF EXPT=RCVD
2204 005764 001410 BEQ LT16X ;IF SO: BR
2205 005766 012737 005746 000706 MOV #LT16IT,SCOLP ;SET LOOP ADDRESS
2206 005774 012737 022251 000666 MOV #MSG27,ERADD ;SET ERROR CODE
2207 006002 004737 016334 JSR PC,LTGER1 ;GO PRINT ERROR
2208 006006 004737 016576 LT16X: JSR PC,ITER ;GO SEE IF ITERATION
2209 006012 000137 002526 LT16XX: JMP TSCD2 ;RETURN TO SCHED
2210 006016 116701 STWD16: .WORD 116701 ;SET UP TEST WORD FOR TE16
2211 ;CONTENTS OF TEST WORD FOR
2212 ;A TU77 IS 116741
2213
    
```

```
2214  
2215 ;LOGIC TEST 17: STATUS AT ON LINE, LOADED*****  
2216  
2217 006020 032777 001000 172542 LT17: BIT #1000,@SWR ;SEE IF INHIB MAN TST  
2218 006026 001005 BNE LT17A ;IF NOT: BR  
2219 006030 005737 000734 TST STFLG ;SEE IF SINGLE TEST  
2220 006034 001433 BEQ LT17XX ;IF NOT: BR  
2221 006036 000137 016646 JMP INMT ;ELSE GO PRINT INHIB MSG  
2222 006042 012737 025110 000616 LT17A: MOV #MSLT17,EMADDR ;SET TEST HEADER  
2223 006050 012704 027204 MOV #MMSG4,R4  
2224 006054 004737 017176 JSR PC,INST ;GO PRINT INSTRUCTION  
2225 006060 004737 016734 LT17IT: JSR PC,INIT2 ;SELECT DRIVE, SLAVE  
2226 006064 013701 006130 MOV STWD17,R1 ;SET TEST WORD  
2227 006070 017702 172426 MOV @DS,R2 ;READ STATUS  
2228 006074 020102 CMP R1,R2 ;SEE IF EXPT=RCVD  
2229 006076 001410 BEQ LT17X ;IF SO: BR  
2230 006100 012737 006060 000706 MOV #LT17IT,SCOLP ;SET LOOP ADDRESS  
2231 006106 012737 022251 000666 MOV #MSG27,ERADD ;SET ERROR CODE  
2232 006114 004737 016334 JSR PC,LTGER1 ;YES PRINT ERROR  
2233 006120 004737 016576 LT17X: JSR PC,ITER ;GO SEE IF ITERATIONS  
2234 006124 000137 002526 LT17XX: JMP TSCD2 ;RETURN TO SCHED  
2235 006130 110701 STWD17: .WORD 110701 ;TEST WORD FOR TE16  
2236 ;CONTENTS OF TEST WORD  
2237 ;IS 110741 FOR A TU77
```

```
2238 ;THE FOLLOWING 11 TESTS WILL TEST ALL POSSIBLE ERROR BITS
2239 ;BY FORCING THEIR CONDITIONS THROUGH VARIOUS ILLEGAL PROGRAMMING
2240 ;SEQUENCES AND USING THE MAINTENANCE WILL MODES AVAILABLE WITH TM03
2241 ;FOR EACH ERROR CONDITION SET THE APPROPRIATE STATUS WILL BE
2242 ;CHECKED. IE: ERR, ATA, SLA, SC ETC.
2243
2244 ;LOGIC TEST 20: ILLEGAL FUNCTION (ILF)*****
2245
2246 006132 012737 025171 000616 LT20: MOV #MSLT20,EMADDR ;SET TEST HEADER
2247 006140 012737 006160 000706 MOV #LT20A,SCOLP ;SET LOOP ADDRESS
2248 006146 012700 000022 LT20IT: MOV #22,R0 ;SET NUMBER OF ILL CODES
2249 006152 012737 000544 000670 MOV #ILFT,TEMP1 ;POINT TO START IF TABLE
2250 006160 004737 016724 LT20A: JSR PC,INIT1 ;GO INIT, SELECT SLAVE + DRIVE
2251 006164 012777 177777 172320 MOV #-1,@WC ;SET WC= -1
2252 006172 012701 000001 MOV #1,R1 ;SET TEST WORD
2253 006176 117777 172466 172304 MOVB @TEMP1,@C1 ;SET ILL CODE
2254 006204 017702 172314 MOV @ER,R2 ;READ ER
2255 006210 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2256 006212 001011 BNE LT20B ;IF SO: BR
2257 006214 012737 027507 000666 MOV #TMS17,ERADD ;SET ERROR CODE
2258 006222 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2259 006230 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2260 006234 000404 BR LT20C
2261 006236 020102 LT20B: CMP R1,R2 ;SEE UNEXPECTED ERRORS
2262 006240 001402 BEQ LT20C ;IF NOT: BR
2263 006242 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
2264 006246 005300 LT20C: DEC R0 ;SEE IF DONE ALL ILL CODES
2265 006250 001403 BEQ LT20X ;IF SO: BR
2266 006252 005237 000670 INC TEMP1 ;BUMP ADDRESS
2267 006256 000740 BR LT20A ;CONTINUE
2268 006260 004737 016576 LT20X: JSR PC,ITER ;GO SEE IF ITERATION
2269 006264 004737 015666 JSR PC,DRVCLR
2270 006270 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```
2271
2272 ;LOGIC TEST 21: REGISTER MODIFICATION REFUSED(RMR)*****
2273
2274 006274 012737 025250 000616 LT21: MOV #MSLT21,EMADDR ;SET TEST HEADER
2275 006302 012737 006310 000706 MOV #LT21IT,SCOLP ;SET SCOPE LOOP ADDRESS
2276 006310 004737 016724 LT21IT: JSR PC,INIT1 ;GO INIT, SELECT SLAVE, DRIVE
2277 006314 052777 000300 172220 BIS #300,@TC ;SET FORMAT
2278 006322 012777 000015 172204 MOV #15,@MR ;SET WAM3
2279 006330 012777 000071 172152 MOV #71,@C1 ;SET READ+GO
2280 006336 005077 172154 CLR @FC ;ATTEMPT WRITE TO FC
2281 006342 012701 000004 MOV #4,R1 ;SET TEST WORD
2282 006346 017702 172152 MOV @ER,R2 ;GET ER
2283 006352 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2284 006354 001011 BNE LT21A ;IF SO: BR
2285 006356 012737 027523 000666 MOV #TMS19,ERADD ;SET ERROR CODE
2286 006364 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2287 006372 004737 015220 JSR PC,LTGER0 ;GO PRINT ERROR
2288 006376 000404 BR LT21B
2289 006400 020102 LT21A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2290 006402 001402 BEQ LT21B ;IF NOT: BR
2291 006404 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2292 ;++B LT21B: JSR PC,ITER ;++B DELETED GO SEE IF ITERATION
2293 006410 012703 040000 LT21B: MOV #40000,R3
2294 006414 005303 LT21XA: DEC R3 ;DELAY FOR ALPHA
2295 006416 001376 BNE LT21XA
2296 006420 004737 015052 JSR PC,EORPA ;GO DO EOR CLEAR
2297 006424 004737 015666 JSR PC,DRVCLR
2298 006430 004737 016576 JSR PC,ITER ;++B GO SEE IF ITERATION
2299 006434 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```
2300
2301
2302
2303 006440 012737 025304 000616 LT22: MOV #MSLT22,EMADDR ;SET TEST HEADER
2304 006446 012737 006454 000706 MOV #LT22IT,SCOLP ;SET SCOPE LOOP ADDRESS
2305 006454 004737 016724 LT22IT: JSR PC,INITI ;INIT, SELECT SLAVE+DRIVE
2306 006460 052777 000020 172032 BIS #20,@CS ;ENABLE EVEN PARITY ON MB
2307 006466 012777 177777 172022 MOV #-1,@FC ;WRITE TO FC
2308 006474 012701 000010 MOV #10,R1 ;SET TEST WORD
2309 006500 042777 000020 172012 BIC #20,@CS ;RESET PARITY TO ODD
2310 006506 017702 172012 MOV @ER,R2 ;GET ER
2311 006512 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2312 006514 001011 BNE LT22A ;IF SO: BR
2313 006516 012737 027531 000666 MOV #TMS20,ERADD ;SET ERROR CODE
2314 006524 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2315 006532 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2316 006536 000404 BR LT22X
2317 006540 020102 LT22A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2318 006542 001402 BEQ LT22X ;IF NOT: BR
2319 006544 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2320 006550 004737 016576 LT22X: JSR PC,ITER ;GO SEE IF ITERATION
2321 006554 004737 015666 JSR PC,DRVCLR
2322 006560 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```
2323
2324
2325
2326 006564 012737 025341 000616 LT23: MOV #MSLT23,EMADDR ;SET TEST HEADER
2327 006572 012737 006600 000706 MOV #LT23IT,SCOLP ;SET SCOPE ADDRESS
2328 006600 004737 016724 LT23IT: JSR PC,INIT1 ;GO INIT SELECT DRIVE+SLAVE
2329 ;++B BIC #360,@TC ;++B DELETED SET ILLEGAL FORMAT
2330 006604 052777 000360 171730 BIS #360,@TC ;++B SET ILLEGAL FORMAT FOR BOTH M8906 & M8915
2331 006612 012701 000020 MOV #20,R1 ;SET TEST WORD
2332 006616 012777 000015 171710 MOV #15,@MR ;SET WAM 3
2333 006624 012777 000071 171656 MOV #71,@C1 ;SET READ+GO
2334 006632 017702 171666 MOV @ER,R2 ;READ ER
2335 006636 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2336 006640 001011 BNE -LT23A ;IF SO: BR
2337 006642 012737 027540 000666 MOV #TMS21,ERADD ;SET ERROR CODE
2338 006650 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2339 006656 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2340 006662 000404 BR LT23X
2341 006664 020102 LT23A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2342 006666 001402 BEQ LT23X ;IF NOT: BR
2343 006670 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2344 006674 004737 016576 LT23X: JSR PC,ITER ;GO SEE IF ITERATION
2345 006700 004737 015052 JSR PC,EORPA
2346 006704 004737 015666 JSR PC,DRVCLR
2347 006710 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```

2348 ;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)*****
2349
2350 006714 012737 025406 000616 LT24: MOV #MSLT24,EMADDR ;SET TEST HEADER
2351 006722 012737 006730 000706 MOV #LT24IT,SCOLP ;SET SCOPE ADDRESS
2352 006730 012737 000005 000602 LT24IT: MOV #5,ITAMT
2353 006736 004737 016752 JSR PC,INIT3 ;GO INIT, SELECT DRIVE+SLAVE
2354 006742 052777 000300 171572 BIS #300,@TC ;SET NORMAL FORMAT
2355 006750 012777 030040 171536 MOV #WDATA,@BA ;SET BA
2356 006756 012777 177760 171532 MOV #-20,@FC ;SET FC
2357 006764 012777 177770 171520 MOV #-10,@WC ;SET WC
2358 006772 012777 000013 171534 MOV #13,@MR ;SELECT WAM 2
2359 007000 012777 000061 171502 MOV #61,@C1 ;SET WRITE+GO
2360 007006 052777 000020 171504 BIS #20,@CS ;FORCE EVEN PARITY
2361 007014 012701 000040 MOV #40,R1 ;SET TEST WORD
2362 007020 012703 000004 MOV #4,R3
2363 007024 005000 CLR R0
2364 007026 005300 1$: DEC R0
2365 007030 001376 BNE 1$ ;DELAY
2366 007032 005303 DEC R3
2367 007034 001374 BNE 1$
2368 007036 012700 000004 MOV #4,R0
2369 007042 012777 000013 171464 LT24B: MOV #13,@MR ;CLOCK MR 4 TIMES
2370 007050 005300 DEC R0
2371 007052 022700 000002 CMP #2,R0 ;SEE IF DONE 1 BYTE
2372 007056 001002 BNE LT24B0 ;IF NOT: BR
2373 007060 017701 171450 MOV @MR,R1 ;ELSE GET BYTE 1
2374 007064 005700 LT24B0: TST R0 ;SEE IF BYTE 2
2375 007066 001365 BNE LT24B ;IF NOT: BR
2376 007070 017704 171440 MOV @MR,R4 ;GET BYTE 2
2377 007074 005000 CLR R0
2378 007076 005300 LT24C: DEC R0
2379 007100 001376 BNE LT24C ;DELAY
2380 007102 032777 000040 171414 BIT #40,@ER ;SEE IF DPAR IS SET
2381 007110 001023 BNE LT24D ;IF SO: BR
2382 007112 000301 SWAB R1
2383 007114 042701 177400 BIC #177400,R1 ;GET LOW BYTE
2384 007120 042704 000377 BIC #377,R4
2385 007124 050401 BIS R4,R1 ;GET HIGH BYTE
2386 007126 005237 000740 INC T24FL ;SET T24 FLAG
2387 007132 012737 027546 000666 MOV #TMS22,ERADD ;SET ERROR CODE
2388 007140 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2389 007146 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2390 007152 005037 000740 CLR T24FL ;CLEAR FLAG
2391 007156 000412 BR LT24X
2392 007160 012701 000050 LT24D: MOV #50,R1
2393 007164 017702 171334 MOV @ER,R2 ;GET ERROR REGISTER
2394 007170 042702 020000 BIC #20000,R2 ;MASK OPI
2395 007174 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2396 007176 001402 BEQ LT24X ;IF NOT: BR
2397 007200 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2398 007204 042777 000020 171306 LT24X: BIC #20,@CS ;RESET EVEN PARITY
2399 007212 004737 015052 JSR PC,EORPA ;GO DO EOR CLEAR
2400 007216 004737 015666 JSR PC,DRVCLR ;GO SEE IF DRIVE CLEAR OK
2401 007222 004737 016576 JSR PC,ITER ;GO SEE IF ITERATION
2402 007226 012737 000020 000602 MOV #20,ITAMT
2403 007234 000137 002526 JMP TSCD2 ;RETURN TO SCHED
  
```

```
2404  
2405 ;LOGIC TEST 25: NON-EXECUTABLE FUNCTION(NEF)*****  
2406  
2407 007240 012737 025446 000616 LT25: MOV #MSLT25,EMADDR ;SET TEST HEADER  
2408 007246 004737 016752 LT25IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE  
2409 007252 052777 000300 171262 BIS #300,@TC ;SET NORMAL FORMAT  
2410 007260 012777 177777 171230 MOV #-1,@FC ;SET ITLLEGAL FC  
2411 007266 012777 000013 171240 MOV #13,@MR ;SET WAM 2  
2412 007274 012777 000061 171206 MOV #61,@C1 ;LOAD WRITE+GO  
2413 007302 012701 004000 MOV #4000,R1 ;SET TEST WORD  
2414 007306 017702 171212 MOV @ER,R2 ;GET ER  
2415 007312 030102 BIT R1,R2 ;SEE IF EXPT=RCVD  
2416 007314 001014 BNE LT25A ;IF SO: BR  
2417 007316 012737 007246 000706 MOV #LT25IT,SCOLP ;SET LOOP ADDRESS  
2418 007324 012737 027634 000666 MOV #TMS31,ERADD ;SET ERROR CODE  
2419 007332 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLAG  
2420 007340 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR  
2421 007344 000404 BR LT25X  
2422 007346 020102 LT25A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS  
2423 007350 001402 BEQ LT25X ;IF NOT: BR  
2424 007352 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR  
2425 007356 004737 016576 LT25X: JSR PC,ITER ;GO SEE IF ITERATION  
2426 007362 004737 015666 JSR PC,DRVCLR  
2427 007366 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```


2428
 2429
 2430
 2431
 2432
 2433
 2434
 2435
 2436
 2437
 2438
 2439
 2440
 2441
 2442
 2443
 2444
 2445
 2446
 2447
 2448
 2449
 2450
 2451
 2452
 2453
 2454
 2455
 2456
 2457
 2458
 2459
 2460
 2461
 2462
 2463
 2464
 2465
 2466
 2467

007372 012737 025502 000616
 007400 004737 016752
 007404 005000
 007406 005300
 007410 001376
 007412 052777 000300 171122
 007420 012777 177770 171064
 007426 012777 177760 171062
 007434 012777 000013 171072
 007442 012777 000061 171040
 007450 012701 001000
 007454 005000
 007456 005300
 007460 001376
 007462 012777 000025 171044
 007470 105077 171040
 007474 012703 000004
 007500 005000
 007502 032777 001000 171014
 007510 001022
 007512 005300
 007514 001372
 007516 005303
 007520 001370
 007522 017702 170776
 007526 012737 007400 000706
 007534 012737 027613 000666
 007542 012737 000001 000712
 007550 004737 015220
 007554 000406
 007556 017702 170742
 007562 020102
 007564 001402
 007566 004737 015206
 007572 004737 016576
 007576 004737 015666
 007602 000137 002526

```

;LOGIC TEST 26: FRAME COUNT ERROR(FCE)*****
LT26:  MOV      #MSLT26,EMADDR ;SET TEST HEADER
LT26IT: JSR      PC,INIT3      ;INIT, SELECT DRIVE+SLAVE
      CLR      RO
1$:    DEC      RO
      BNE      1$              ;AWAIT OPI RESET
      BIS      #300,@TC        ;SET NORMAL FORMAT
      MOV      #-10,@WC        ;SET WC=-10
      MOV      #-20,@FC        ;SET FC=-20
      MOV      #13,@MR         ;SET WAM 3
      MOV      #61,@C1         ;LOAD WRITE+GO
      MOV      #1000,R1        ;SET TEST WORD
      CLR      RO
2$:    DEC      RO
      BNE      2$              ;DELAY
      MOV      #25,@MR         ;LOAD MM EOR CLEAR
      CLRB     @MR             ;RESET MR
      MOV      #4,R3
      CLR      RO
3$:    BIT      #1000,@ER       ;SEE IF FCE SET
      BNE      4$              ;IF SO: BR
      DEC      RO
      BNE      3$              ;DELAY
      DEC      R3
      BNE      3$
      MOV      @ER,R2          ;GET ER
      MOV      #LT26IT,SCOLP   ;SET SCOPE ADDRESS
      MOV      #TMS28,ERADD
      MOV      #1,EXFL         ;SET EXPT FLG
      JSR      PC,LTGERO       ;GO PRINT ERROR
      BR      LT26X
4$:    MOV      @ER,R2          ;GET ERROR REGISTER
      CMP      R1,R2           ;SEE IF UNEXPECTED ERRORS
      BEQ      LT26X           ;IF NOT: BR
      JSR      PC,LTGER3       ;ELSE GO PRINT ERROR
LT26X: JSR      PC,ITER         ;GO SEE IF ITERATION
      JSR      PC,DRVCLR
      JMP      TSCD2           ;RETURN TO SCHED
  
```

```

2468
2469                                     ;LOGIC TEST 27: ILLEGAL REGISTER(ILR)*****
2470
2471 007606 022737 172400 000510 LT27:  CMP      #172400,C1      ;SEE IF ADDRESSES OPEN
2472 007614 001041                                     BNE      LT27XX      ;IF NOT: BR
2473 007616 012737 007642 000706          MOV      #LT27A,SCOLP ;SET SCOPE ADDRESS
2474 007624 012737 025536 000616          MOV      #MSLT27,EMADDR ;SET TEST HEADER
2475 007632 012700 000020          LT27IT: MOV      #20,R0      ;SET NUMBER OF ILR TESTS
2476 007636 012701 172434          MOV      #172434,R1   ;SET FIRST ILR ADDRESS
2477 007642 004737 016752          LT27A:  JSR      PC,INIT3   ;GO INIT, SELECT DRIVE+SLAVE
2478 007646 011103          MOV      (R1),R3      ;ATTEMPT ILR READ
2479 007650 032777 000002 170646          BIT      #2,@ER      ;SEE IF ILR=1
2480 007656 001010          BNE      LT27B      ;IF SO: BR
2481 007660 012737 000001 000712          MOV      #1,EXFL     ;SET EXPT-NOT RCVD FLAG
2482 007666 012737 027435 000666          MOV      #TMS10,ERADD ;SET ERROR CODE
2483 007674 004737 015226          JSR      PC,LTGER    ;GO PRINT ERROR
2484 007700 005300          LT27B:  DEC      R0      ;SEE IF DONE ALL
2485 007702 001402          BEQ      LT27X      ;IF SO: BR
2486 007704 005721          TST      (R1)+      ;BUMP ADDRESS
2487 007706 000755          BR      LT27A      ;CONTINUE TESTS
2488 007710 004737 016576          LT27X:  JSR      PC,ITER   ;GO SEE IF ITERATIONS
2489 007714 004737 015666          JSR      PC,DRVCLR
2490 007720 000137 002526          LT27XX: JMP      TSCD2    ;RETURN TO SCHED
  
```

```

2491
2492
2493 ;LOGIC TEST 30: DRIVE TIMING ERROR*****
2494 007724 012737 027642 000666 LT30: MOV #TMS32,ERADD ;SET ERROR CODE
2495 007732 012737 025572 000616 MOV #MSLT30,EMADDR ;SET TEST HEADER
2496 007740 012737 007746 000706 MOV #LT30IT,SCOLP ;SET SCOPE ADDRESS
2497 007746 004737 016752 LT30IT: JSR PC,INIT3 ;INIT, SELECT DRIVE + SLAVE
2498 007752 052777 000300 170562 BIS #300,@TC ;SET NORMAL FORMAT
2499 007760 012701 010000 MOV #10000,R1 ;SET TEST WORD
2500 007764 012777 000017 170542 MOV #17,@MR ;CRIPPLE OCCUPIED
2501 007772 005077 170520 CLR @FC ;SET FC3
2502 007776 012777 000061 170504 MOV #61,@C1 ;LOAD WRITE+GO
2503 010004 032777 010000 170512 BIT #10000,@ER ;SEE IF DTE SET
2504 010012 001005 LT30A: BNE LT30A ;IF SO: BR
2505 010014 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2506 010022 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2507 010026 004737 016752 LT30A: JSR PC,INIT3 ;GO INIT SELECT DRIVE,SLAVE
2508 010032 052777 000300 170502 BIS #300,@TC ;SET FORMAT
2509 010040 012701 010000 MOV #10000,R1 ;SET TEST WORD
2510 010044 005077 170446 CLR @FC ;SET FCS
2511 010050 012777 000015 170456 MOV #15,@MR ;SET WRAP 3
2512 010056 012777 000061 170424 MOV #61,@C1 ;LOAD WRITE+GO
2513 010064 012704 040000 MOV #40000,R4
2514 010070 005777 170446 LT30B: TST @TC ;SEE IF ALPHA
2515 010074 100015 BPL LT30C ;AWAIT ALPHA
2516 010076 005300 DEC R0
2517 010100 001373 BNE LT30B
2518 010102 013704 000616 MOV EMADDR,R4
2519 010106 004737 017724 JSR PC,TTOUT ;PRINT HEADER
2520 010112 012704 023005 MOV #MSG50,R4
2521 010116 004737 017724 JSR PC,TTOUT ;PRINT ALPHA ERROR
2522 010122 004737 016546 JSR PC,SCOPE
2523 010126 000435 BR LT30X
2524 010130 012777 000015 170376 LT30C: MOV #15,@MR ;CLOCK MR
2525 010136 012777 000015 170370 MOV #15,@MR ;CLOCK MR
2526 010144 005000 CLR R0
2527 010146 005300 LT30D: DEC R0
2528 010150 001376 BNE LT30D ;DELAY
2529 010152 032777 010000 170344 BIT #10000,@ER ;SEE IF DTE SET
2530 010160 001006 BNE LT30E ;IF SO: BR
2531 010162 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2532 010170 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2533 010174 000412 BR LT30X
2534 010176 012701 010000 LT30E: MOV #10000,R1 ;SET TEST WORD
2535 010202 017702 170316 MOV @ER,R2 ;GET ERROR REGISTER
2536 010206 042702 020100 BIC #20100,R2 ;MASK OPI AND VPE
2537 010212 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2538 010214 001402 BEQ LT30X ;IF NOT: BR
2539 010216 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2540 010222 004737 016576 LT30X: JSR PC,ITER ;GO SEE IF ITERATION
2541 010226 004737 015052 JSR PC,EORPA ;GO CLEAR GO BIT
2542 010232 004737 015666 JSR PC,DRVCLR
2543 010236 000137 002526 JMP TSCD2 ;RETURN TO SCHED
2544

```

```

2545
2546 ;LOGIC TEST 31: OPERATION INCOMPLETE(OPI)*****
2547
2548 010242 012737 025630 000616 LT31: MOV #MSLT31,EMADDR ;SET TEST HEADER
2549 010250 012737 010250 000706 LT31IT: MOV #LT31IT,SCOLP ;SET SCOPE ADDRESS
2550 010256 012737 027656 000666 MOV #TMS33A,ERADD ;SET ERROR MSG HDR
2551 010264 012737 000002 000602 MOV #2,ITAMT ;SET REDUCED ITER COUNT
2552 010272 004737 016752 JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2553 010276 005000 CLR R0
2554 010300 005300 1$: DEC R0
2555 010302 001376 BNE 1$ ;AWAIT OPI RESET
2556 010304 052777 000300 170230 BIS #300,@TC ;SET FORMAT
2557 010312 012777 000013 170214 MOV #13,@MR ;SET WAM 2
2558 010320 005077 170172 CLR @FC ;SET FRAME COUNT
2559 010324 012705 020000 MOV #20000,R5 ;SET TEST BIT (OPI)
2560 010330 012702 010346 MOV #2$,R2 ;SET RETURN ADDRESS FROM TIMER
2561 010334 004737 010546 JSR PC,TIMON ;START TIMER
2562 010340 012777 000061 170142 MOV #61,@C1 ;LOAD WRITE+GO
2563 010346 030577 170152 2$: BIT R5,@ER ;BRANCH WHEN OPI SETS
2564 010352 001002 BNE 3$
2565 010354 000163 010640 JMP TIMER(R3) ;GO TO TIMER & RETURN TO 2$ ABOVE
2566 010360 017702 170140 3$: MOV @ER,R2 ;GET ERROR REGISTER
2567 010364 020502 CMP R5,R2 ;SEE IF UNEXPECTED ERRORS
2568 010366 001403 BEQ 4$ ;IF NOT: BR
2569 010370 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
2570 010374 000453 BR LT31X
2571 010376 004737 010732 4$: JSR PC,TIMOK ;GO CHECK TIME FOR OPI TO SET
2572 010402 102450 BVS LT31X ;BRANCH IF TIME WAS INCORRECT
2573
2574 010404 012737 010420 000706 MOV #LT31A,SCOLP ;SET SCOPE LOOP
2575 010412 012737 027672 000666 MOV #TMS33B,ERADD ;SET ERROR MSG HEADER
2576 010420 004737 016752 LT31A: JSR PC,INIT3 ;GO INIT
2577 010424 005000 CLR R0
2578 010426 005300 1$: DEC R0 ;WAIT FOR OPI TO CLEAR
2579 010430 001376 BNE 1$
2580 010432 052777 000300 170102 BIS #300,@TC ;SET FORMAT
2581 010440 012777 000015 170066 MOV #15,@MR ;SET WRAP 3
2582 010446 012702 010470 MOV #2$,R2 ;SET RETURN ADDRESS FROM TIMER
2583 010452 012705 020000 MOV #20000,R5 ;SET TEST WORD
2584 010456 004737 010546 JSR PC,TIMON ;START TIMER
2585 010462 012777 000071 170020 MOV #71,@C1 ;LOAD READ+GO
2586 010470 030577 170030 2$: BIT R5,@ER ;BRANCH WHEN OPI SETS
2587 010474 001002 BNE 3$
2588 010476 000163 010640 JMP TIMER(R3) ;GO TO TIMER
2589 010502 017702 170016 3$: MOV @ER,R2 ;GET ERROR REGISTER
2590 010506 020502 CMP R5,R2 ;SEE IF UNEXPECTED ERRORS
2591 010510 001403 BEQ 4$
2592 010512 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
2593 010516 000402 BR LT31X ;EXIT TEST
2594 010520 004737 010732 4$: JSR PC,TIMOK ;GO CHECK TIME
2595 010524 004737 016576 LT31X: JSR PC,ITER ;GO SEE IF ITERATIONS
2596 010530 004737 015666 JSR PC,DRVCLR
2597 010534 012737 000020 000602 MOV #20,ITAMT
2598 010542 000137 002526 JMP TSCD2 ;RETURN TO SCHED
2599
2600

```

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

```

2601 ;ENANCE REGISTER (BIT 6) THAT TOGGLES EVERY 56 (10) MICROSECONDS. THIS
2602 ;ROUTINE WAITS FOR THE OSCILLATOR TO TOGGLE AND RETURN WITH R3 INDICATING
2603 ;THE STATE OF THE OSCILLATOR.
2604 010546 005000 TIMON: CLR R0 ;CLEAR TICK COUNT
2605 010550 005001 CLR R1
2606 010552 012703 000024 MOV #24,R3 ;PRESET INDEX TO TIMER
2607 010556 032777 000100 167750 BIT #100,@MR ;BRANCH IF OSC CLEAR
2608 010564 001405 BEQ 2$
2609 010566 032777 000100 167740 1$: BIT #100,@MR ;WAIT FOR OSC TO CLEAR
2610 010574 001374 BNE 1$
2611 010576 000405 BR 4$ ;EXIT
2612
2613 010600 005403 2$: NEG R3 ;SET INDEX TO TIMER
2614 010602 032777 000100 167724 3$: BIT #100,@MR ;WAIT FOR OSC TO SET
2615 010610 001774 BEQ 3$
2616 010612 000207 4$: RTS PC ;RETURN
2617
2618 ;THIS ROUTINE TIMES AN EVENT. EACH TIME THE OSCILLATOR BIT CHANGES
2619 ;STATE THE TICK COUNT IN R1 & R0 IS INCREMENTED. THE ROUTINE IS CALLED
2620 ;USING R3 AS AN INDEX TO INDICATE THE OSCILLATORS PAST STATE. WHEN
2621 ;THE OSC BIT CHANGES STATE R3 IS NEGATED.
2622 010614 032777 000100 167712 TIMER1: BIT #100,@MR ;BRANCH IF OSC HAS CHANGED STATE
2623 010622 001406 BEQ TIMER
2624 010624 000112 JMP (R2) ;RETURN
2625 010640 010640 .=TIMER1+24
2626 010640 005403 TIMER: NEG R3 ;SET INDEX TO OTHER STATE
2627 010642 062700 000001 ADD #1,R0 ;INCREMENT TICK COUNT
2628 010646 005501 ADC R1
2629 010650 022701 000003 CMP #3,R1 ;BRANCH IF TIMER OVERFLOWS
2630 010654 001410 BEQ TIMOVF
2631 010656 000112 JMP (R2) ;RETURN
2632 010664 010664 .=TIMER+24
2633 010664 032777 000100 167642 TIMER0: BIT #100,@MR ;BRANCH IF OSC SET
2634 010672 001362 BNE TIMER
2635 010674 000112 JMP (R2) ;RETURN
2636
2637 010676 013704 000616 TIMOVF: MOV EMADDR,R4 ;TYPE TEST HEADER
2638 010702 004737 017724 JSR PC,TTOUT
2639 010706 013704 000666 MOV ERADD,R4 ;GET ERROR MSG ADDRESS
2640 010712 004737 017724 JSR PC,TTOUT ;AND TYPE IT
2641 010716 012704 027752 MOV #TMS33E,R4 ;TYPE
2642 010722 004737 017724 JSR PC,TTOUT ;'TIMER OVERFLOWED'
2643 010726 000137 010524 JMP LT31X ;GO EXIT TEST
2644
2645 ;ROUTINE TO CHECK IF TIME IS WITHIN LIMITS. IF NOT THE ROUTINE RETURNS
2646 ;WITH THE 'V' BIT SET. THE LIMITS WERE SLECTED BY DIVIDING THE TIME
2647 ;IN MICROSECONDS BY 448. THE LOWER LIMIT IS 5,500,000 USECS (5.5 SECS);
2648 ;THE UPPER LIMIT IS 9,500,000 USECS (9.5 SECS). THE 448 IS DERIVED FROM
2649 ;56 USECS/TICK TIMES THE DIVISION BY 8 BY THE TIMOK ROUTINE.
2650 010732 000240 TIMOK: NOP
2651 010734 006201 ASR R1 ;DIVIDE COUNT BY 8
2652 010736 006000 ROR R0
2653 010740 006201 ASR R1
2654 010742 006000 ROR R0
2655 010744 006201 ASR R1
2656 010746 006000 ROR R0

```

```

2657 010750 013701 001010      MOV      SLVTYP,R1      ;++B GET SLAVE TYPE (0/1 = TE16/TU77)
2658 010754 006301              ASL      R1              ;++B FORM INDEX
2659 010756 020061 011062      CMP      RO,200$(R1)    ;++B BRANCH IF GREATER THAN LOWER LIMIT(5.5 SECS)
2660 010762 101016              BHI     1$
2661 010764 013704 000616      MOV      EMADDR,R4     ;GET ERROR MSG HEADER
2662 010770 004737 017724      JSR     PC,TTOUT       ;TYPE ERROR MSG HEADER
2663 010774 013704 000666      MOV      ERADD,R4     ;GET ERROR DESCRIPTOR MSG
2664 011000 004737 017724      JSR     PC,TTOUT
2665 011004 012704 027705      MOV      #TMS33C,R4   ;TYPE 'OCCURED TOO SOON'
2666 011010 004737 017724      JSR     PC,TTOUT
2667 011014 000262              SEV
2668 011016 000420              BR      2$
2669
2670 011020 020061 011066      1$:     CMP      RO,201$(R1) ;++B BRANCH IF LESS THAN UPPER LIMIT(9.5 SECS)
2671 011024 003415              BLE     2$
2672 011026 013704 000616      MOV      EMADDR,R4     ;GET ERROR MSG HEADER
2673 011032 004737 017724      JSR     PC,TTOUT
2674 011036 013704 000666      MOV      ERADD,R4
2675 011042 004737 017724      JSR     PC,TTOUT       ;TYPE ERROR MSG HEADER
2676 011046 012704 027727      MOV      #TMS33D,R4   ;TYPE 'OCCURED TOO LATE'
2677 011052 004737 017724      JSR     PC,TTOUT
2678 011056 000262              SEV
2679 011060 000207      2$:     RTS      PC
2680
2681
2682      ;++B TABLE OF MIN AND MAX TIMES FOR OPI FOR TE16 AND TU77 SLAVES
2683      ;++B MIN TIMES (5.5 SECS)
2684 011062 027764      200$:   .WORD   12276.    ;++B TE16
2685 011064 020622      .WORD   8594.         ;++B TU77
2686
2687      ;++B MAX TIMES (9.5 SECS)
2688 011066 051325      201$:   .WORD   21205.    ;++B TE16
2689 011070 034774      .WORD   14844.       ;++B TU77
  
```

```
2690
2691
2692
2693
2694
2695 011072 012737 025664 000616 LT32: MOV #MSLT32,EMADDR ;SET TEST HEADER
2696 011100 012737 011106 000706 MOV #LT32IT,SCOLP ;SET SCOPE ADDRESS
2697 011106 004737 016752 LT32IT: JSR PC,INIT3 ;INIT, SELECT DRIVE +SLAVE
2698 011112 013700 000660 MOV SLVN,R0 ;GET SLAVE NUMBER
2699 011116 012701 040000 MOV #40000,R1 ;++B SET TEST WORD (UNS)
2700 011122 032777 000004 167406 BIT #4,@DT ;++B BRANCH IF TE16
2701 011130 001402 BEQ 1$ ;++B
2702 011132 052701 004000 BIS #4000,R1 ;++B SET ALSO NEF FOR TU77
2703 011136 005100 1$: COM R0 ;SET NONEXISTANT SLAVE
2704 011140 042700 177770 BIC #177770,R0 ;MASK SLAVE NUMBER
2705 011144 052700 000300 BIS #300,R0 ;SET FORMAT
2706 011150 010077 167366 MOV R0,@TC ;SELECT ILLEGAL SLAVE
2707 011154 032777 002000 167354 BIT #2000,@DT ;EXIT TEST IF SALVE AVAILABLE
2708 011162 001030 BNE LT32XX
2709 011164 012777 000071 167316 MOV #71,@C1 ;LOAD READ+GO
2710 011172 017702 167326 MOV @ER,R2 ;READ ER
2711 011176 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2712 011200 001011 BNE 2$ ;IF SO: BR
2713 011202 012737 027772 000666 MOV #TMS34,ERADD ;SET ERROR CODE
2714 011210 012737 000001 000712 MOV #1,EXFL ;SET ERROR CODE
2715 011216 004737 015220 JSR PC,LTGER0 ;GO PRINT ERROR
2716 011222 000404 BR LT32X
2717 011224 020102 2$: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2718 011226 001402 BEQ LT32X ;IF NOT: BR
2719 011230 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
2720 011234 004737 016576 LT32X: JSR PC,ITER ;GO SEE IF ITERATIONS
2721 011240 004737 015666 JSR PC,DRVCLR
2722 011244 000137 002526 LT32XX: JMP TSCD2 ;RETURN TO SCHED
```

2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743

011250 012737 025720 000616
011256 012737 011264 000706
011264 004737 016752
011270 012777 000013 167236
011276 012777 177777 167212
011304 012777 000031 167176
011312 032777 020000 167202
011320 001010
011322 012737 027465 000666
011330 012737 000001 000712
011336 004737 015220
011342 004737 016576
011346 000137 002526

; THE FOLLOWING 6 TESTS WILL LOOK AT VARIOUS BITS IN THE
; DRIVE STATUS(DS) AND TAPE CONTROL(TC)
; REGISTERS BY FORCING CERTAIN CONDITIONS WHICH DO NOT
; REQUIRE TAPE MOVEMENT.
; LOGIC TEST 33: POSITIONING IN PROGRESS(PIP)*****
LT33: MOV #MSLT33,EMADDR ;SET TEST HEADER
MOV #LT33IT,SCOLP ;SET SCOPE ADDRESS
LT33IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
MOV #13,@MR ;SET WAM 2
MOV #-1,@FC ;SET FCS
MOV #31,@C1 ;LOAD SPACE FORWARD+GO
BIT #20000,@DS ;SEE IF PIP=1
BNE LT33X ;IF SO: BR
MOV #TMS14,ERADD ;SET ERROR CODE
MOV #1,EXFL ;SET ERROR CODE
JSR PC,LTGERO ;GO PRINT ERROR
LT33X: JSR PC,ITER ;GO SEE IF ITERATIONS
JMP TSCD2 ;RETURN TO SCHED


```
2744  
2745  
2746  
2747 011352 012737 027405 000666 LT34: MOV #TMS6,ERADD ;SET ERROR CODE  
2748 011360 012737 025754 000616 MOV #MSLT34,EMADDR ;SET TEST HEADER  
2749 011366 012700 000004 LT34IT: MOV #4,R0  
2750 011372 004737 016752 LT34A1: JSR PC,INIT3 ;GO INIT, SELECT DRIVE+SLAVE  
2751 011376 042777 003400 167136 BIC #3400,@TC ;SELECT NRZI  
2752 011404 052777 001400 167130 BIS #1400,@TC  
2753 011412 032777 000040 167102 LT34A: BIT #40,@DS ;SEE IF PES=0  
2754 011420 001410 BEQ LT34B ;IF SO: BR  
2755 011422 012737 000002 000712 MOV #2,EXFL ;SET RCVD-NOT EXPT  
2756 011430 012737 011372 000706 MOV #LT34A1,SCOLP ;SET SCOPE ADDRESS  
2757 011436 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR  
2758 011442 004737 016766 LT34B: JSR PC,INIT4  
2759 011446 032777 000040 167046 LT34C: BIT #40,@DS ;SEE IF PES=1  
2760 011454 001010 BNE LT34X ;IF SO: BR  
2761 011456 012737 011446 000706 MOV #LT34C,SCOLP ;SET SCOPE ADDRESS  
2762 011464 012737 000001 000712 MOV #1,EXFL ;SET EXPT-NOT RCVD FLAG  
2763 011472 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR  
2764 011476 004737 016576 LT34X: JSR PC,ITER ;GO SEE IF ITERATION  
2765 011502 000137 002526 LT34XX: JMP TSCD2 ;RETURN TO SCHED
```

```
2766
2767
2768
2769 011506 012737 030015 000666 LT35: MOV #TMS37,ERADD
2770 011514 012737 026010 000616 MOV #MSLT35,EMADDR
2771 011522 004737 016752 LT35IT: JSR PC,INIT3 ;INIT SELECT DRIVE, SLAVE
2772 011526 032777 000020 166766 1$: BIT #20,ADS ;SEE IF SDWN IS RESET
2773 011534 001374 BNE 1$ ;IF NOT: BR
2774 011536 052777 000300 166776 BIS #300,@TC ;SET FORMAT
2775 011544 012777 000015 166762 MOV #15,@MR ;SET WAM 3
2776 011552 012777 000071 166730 MOV #71,@C1 ;LOAD READ+GO
2777 011560 032777 020000 166754 BIT #20000,@TC ;SEE IF SAC=0
2778 011566 001410 BEQ LT35A ;IF SO: BR
2779 011570 012737 000002 000712 MOV #2,EXFL ;SET RCV-NOT EXPT FLAG
2780 011576 012737 011522 000706 MOV #LT35IT,SCOLP ;SET SCOPE ADDRESS
2781 011604 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2782 011610 004737 016752 LT35A: JSR PC,INIT3 ;INIT
2783 011614 005277 166722 INC @TC ;BUMP SLAVE ADDRESS
2784 011620 032777 020000 166714 BIT #20000,@TC ;SEE IF SAC=1
2785 011626 001010 BNE LT35X ;IF SO: BR
2786 011630 012737 011610 000706 MOV #LT35A,SCOLP ;SET SCOPE ADDRESS
2787 011636 012737 000001 000712 MOV #1,EXFL ;SE EXPT-NOT RCVD FLAG
2788 011644 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2789 011650 004737 016576 LT35X: JSR PC,ITER
2790 011654 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```
2791
2792
2793
2794 011660 012737 026055 000616 LT36: MOV #MSLT36,EMADDR
2795 011666 012737 030023 000666 MOV #TMS38,ERADD ;SET ERROR CODE
2796 011674 004737 016752 LT36IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2797 011700 032777 040000 166634 BIT #40000,@TC ;SEE IF FCS=0
2798 011706 001410 BEQ 1$ ;IF SO: BR
2799 011710 012737 011674 000706 MOV #LT36IT,SCOLP ;SET SCOPE ADDRESS
2800 011716 012737 000002 000712 MOV #2,EXFL ;SET RCVD-NOT EXPT
2801 011724 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2802 011730 004737 016752 1$: JSR PC,INIT3 ;INIT
2803 011734 005077 166556 CLR @FC ;WRITE TO FC
2804 011740 032777 040000 166574 BIT #40000,@TC ;SEE IF FCS=1
2805 011746 001010 BNE LT36X ;IF SO: BR
2806 011750 012737 011730 000706 MOV #1$,SCOLP ;SET SCOPE ADDRESS
2807 011756 012737 000001 000712 MOV #1,EXFL ;SET EXPT-NOT RCVD
2808 011764 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2809 011770 004737 016576 LT36X: JSR PC,ITER
2810 011774 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```
2811
2812
2813
2814 012000 012737 026122 000616 LT37: MOV #MSLT37,EMADDR
2815 012006 012737 030031 000666 MOV #TMS39,ERADD ;SET ERROR CODE
2816 012014 004737 016752 LT37IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2817 012020 052777 000300 166514 BIS #300,@TC ;SET FORMAT
2818 012026 005777 166510 TST @TC ;SEE IF ACCL=1
2819 012032 100410 BMI LT37A ;IF SO: BR
2820 012034 012737 000001 000712 MOV #1,EXFL
2821 012042 012737 012014 000706 MOV #LT37IT,SCOLP ;SET SCOPE ADDRESS
2822 012050 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2823 012054 004737 016752 LT37A: JSR PC,INIT3 ;INIT
2824 012060 052777 000300 166454 BIS #300,@TC ;SET FORMAT
2825 012066 012777 000015 166440 MOV #15,@MR ;SET WAM 3
2826 012074 012777 000071 166406 MOV #71,@C1 ;LOAD READ+GO
2827 012102 012700 100000 MOV #100000,RO ;SET ACCL DELAY
2828 012106 005777 166430 LT37B: TST @TC ;SEE IF ACCL=0
2829 012112 100012 BPL LT37X ;IF SO: BR
2830 012114 005300 DEC RO
2831 012116 001373 BNE LT37B ;DELAY
2832 012120 012737 012054 000706 MOV #LT37A,SCOLP ;SET SCOPE ADDRESS
2833 012126 012737 000002 000712 MOV #2,EXFL
2834 012134 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2835 012140 004737 016576 LT37X: JSR PC,ITER
2836 012144 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```
2837  
2838  
2839  
2840 012150 012737 012164 000706 LT40: MOV #LT40IT,SCOLP ;SET SCOPE ADDRESS  
2841 012156 012737 026170 000616 MOV #MSLT40,EMADDR  
2842 012164 004737 016766 LT40IT: JSR PC,INIT4 ;INIT, SELECT DRIVE+SLAVE  
2843 012170 005000 CLR R0  
2844 012172 005300 1$: DEC R0  
2845 012174 001376 BNE 1$ ;DELAY FOR OPI RESET  
2846 012176 052777 002300 166336 BIS #2300,@TC  
2847 012204 012777 000007 166322 MOV #7,@MR ;SET WAM 0  
2848 012212 012777 000027 166270 MOV #27,@C1 ;LOAD WRITE TAPE MARK+GO  
2849 012220 012700 100000 MOV #100000,R0 ;SET DELAY  
2850 012224 032777 000004 166270 2$: BIT #4,@DS ;SEE IF TM=1  
2851 012232 001012 BNE LT40X ;IF SO: BR  
2852 012234 005300 DEC R0  
2853 012236 001372 BNE 2$ ;DELAY  
2854 012240 012737 027363 000666 MOV #TMS3,ERADD  
2855 012246 012737 000001 000712 MOV #1,EXFL  
2856 012254 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR  
2857 012260 004737 016576 LT40X: JSR PC,ITER  
2858 012264 000137 002526 LT40XX: JMP TSCD2 ;RETURN TO SCHED
```

```

2859
2860                               ;LOGIC TEST 41: NRZ TAPE MARK (TM,VPE,ITM)*****
2861
2862 012270 012737 012304 000706 LT41: MOV #LT41IT,SCOLP ;SET SCOPE ADDRESS
2863 012276 012737 026235 000616      MOV #MSLT41,EMADDR
2864 012304 004737 016752          LT41IT: JSR PC,INIT3 ;INIT, SELECT DRIVE,SLAVE
2865 012310 052777 001700 166224      BIS #1700,@TC ;SET NRZ+NORMAL FORMAT
2866 012316 012777 177760 166172      MOV #-20,@FC ;SET FCS
2867 012324 012777 000007 166202      MOV #7,@MR ;SET WAM 0
2868 012332 012777 000027 166150      MOV #27,@C1 ;LOAD WRITE TAPE MARK+GO
2869 012340 005000
2870 012342 032777 000004 166152 1$:  CLR R0
2871 012350 001012          BIT #4,@DS ;SEE IF TM=1
2872 012352 005300          BNE 2$ ;IF SO: BR
2873 012354 001372          DEC R0
2874 012356 012737 027363 000666      BNE 1$ ;DELAY
2875 012364 012737 000001 000712      MOV #TMS3,ERADD ;SET ERROR CODE
2876 012372 004737 015220          MOV #1,EXFL
2877 012376 032777 002000 166120 2$:  JSR PC,LTGERO ;GO PRINT ERROR
2878 012404 001010          BIT #2000,@ER ;SEE IF ITM=1
2879 012406 012737 027626 000666      BNE 3$ ;IF SO: BR
2880 012414 012737 000001 000712      MOV #TMS30,ERADD ;SET ERROR CODE
2881 012422 004737 015220          MOV #1,EXFL
2882 012426 032777 000100 166070 3$:  JSR PC,LTGERO ;GO PRINT ERROR
2883 012434 001011          BIT #100,@ER ;SEE IF VPE=1
2884 012436 012737 027613 000666      BNE 4$ ;IF SO: BR
2885 012444 012737 000001 000712      MOV #TMS28,ERADD ;SET ERROR CODE
2886 012452 004737 015220          MOV #1,EXFL
2887 012456 000410          JSR PC,LTGERO ;GO PRINT ERROR
2888 012460 012701 002100          BR LT41X
2889 012464 017702 166034          4$:  MOV #2100,R1 ;SET EXPT ERROR BITS
2890 012470 020102          MOV @ER,R2 ;GET ERROR REGISTER
2891 012472 001402          CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2892 012474 004737 015206          BEQ LT41X ;IF NOT: BR
2893 012500 005002          JSR PC,LTGER3 ;ELSE PRINT ERROR
2894 012502 032777 000200 166012 1$:  CLR R2 ;SET TIMER
2895 012510 001002          1$:  BIT #200,@DS ;SEE IF DRY SET
2896 012512 005302          BNE 2$ ;IF SO: BR
2897 012514 001372          DEC R2 ;AWAIT DRY
2898 012516 004737 016576          2$:  BNE 1$ ;DELAY
2899 012522 004737 015666          JSR PC,ITER ;GO SEE IF ITERATIONS
2900 012526 000137 002526          JSR PC,DRVCLR ;GO DO DRIVE CLEAR
      JMP TSCD2 ;RETURN TO SCHED
  
```

```

2901
2902
2903
2904
2905
2906
2907
2908 012532 012737 001700 000772 LT42: MOV #1700,UDES ;SET UNIT DESCRIPTION = NRZ
2909 012540 004737 015022 JSR PC,STATIC ;GO SEE IF STATIC ONLY
2910 012544 012700 001000 MOV #1000,RO
2911 012550 005300 1$: DEC RO
2912 012552 001376 BNE 1$ ;PAUSE
2913 012554 012737 026304 000616 MOV #MSLT42,EMADDR
2914 012562 012737 012570 000706 MOV #LT42IT,SCOLP ;SET SCOPE ADDRESS
2915 012570 004737 017002 LT42IT: JSR PC,INIT ;INIT SELECT DRIVE+SLAVE
2916 012574 012777 177770 165710 MOV #-10,@WC
2917 012602 012777 177760 165706 MOV #-20,@FC ;SET FC=20
2918 012610 012777 030040 165676 MOV #WDATA,@BA ;SET BUS ADDRESS
2919 012616 012777 000007 165710 MOV #7,@MR ;SET MM CODE
2920 012624 012777 000061 165656 MOV #61,@C1 ;LOAD WRITE+GO
2921 012632 005000 CLR RO
2922 012634 032777 000200 165660 LT42A: BIT #200,@DS ;SEE IF DRY=1
2923 012642 001002 BNE LT42B ;IF SO: BR
2924 012644 005300 DEC RO
2925 012646 001372 BNE LT42A ;DELAY
2926 012650 022777 000200 165646 LT42B: CMP #200,@ER ;SEE IF LRC ERROR ONLY
2927 012656 001007 BNE LT42B1 ;IF NOT: BR
2928 012660 017702 165644 MOV @CC,R2 ;GET CHECK CHAR
2929 012664 042702 177000 BIC #177000,R2 ;MASK CRC
2930 012670 022702 000777 CMP #777,R2 ;SEE IF SETUP CRC IS CORRECT
2931 012674 001410 BEQ LT42B2 ;IF SO: BR
2932 012676 004737 015206 LT42B1: JSR PC,LTGER3 ;ELSE PRINT ERROR SETUP
2933 012702 012704 023076 MOV #MSG55,R4
2934 012706 004737 017724 JSR PC,TTOUT ;PRINT SETUP ERROR MSG
2935 012712 000137 002526 JMP TSCD2 ;RETURN TO SCHED
2936 012716 004737 017002 LT42B2: JSR PC,INIT ;GO INIT
2937 012722 012777 177770 165562 MOV #-10,@WC ;SET WC
2938 012730 012777 177760 165560 MOV #-20,@FC ;SET FC
2939 012736 012777 030040 165550 MOV #WDATA,@BA ;SET BA
2940 012744 012777 000021 165562 MOV #21,@MR ;SET MM
2941 012752 012777 000061 165530 MOV #61,@C1 ;LOAD WRITE+GO
2942 012760 005000 CLR RO
2943 012762 032777 000200 165532 LT42C: BIT #200,@DS ;SEE IF DRY
2944 012770 001002 BNE LT42D ;IF SO: BR
2945 012772 005300 DEC RO
2946 012774 001372 BNE LT42C ;AWAIT DRY
2947 012776 005777 165522 LT42D: TST @ER ;SEE IF CRC=1
2948 013002 100411 BMI LT42E ;IF SO: BR
2949 013004 012737 030007 000666 MOV #TMS36,ERADD ;SET ERROR CODE
2950 013012 012737 000001 000712 MOV #1,EXFL
2951 013020 004737 015220 JSR PC,LTGER0 ;GO PRINT ERROR
2952 013024 000410 BR LT42X
2953 013026 012701 100200 LT42E: MOV #100200,R1 ;SET EXPT ERROR BITS
2954 013032 017702 165466 MOV @ER,R2 ;GET ERROR REGISTER
2955 013036 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2956 013040 001402 BEQ LT42X ;IF NOT: BR
  
```

```

2957 013042 004737 015206
2958 013046 004737 016576
2959 013052 004737 015666
2960 013056 000137 002526
2961
2962
2963
2964 013062 032777 000004 165446 LT43: BIT #4,@DT ;++B BRANCH IF NOT A TE16
2965 013070 001114 LT43XX ;++B
2966 013072 012737 001700 000772 MOV #1700,UDES ;SET UNIT DESCRIPTION = NRZ
2967 013100 004737 015022 JSR PC,STATIC ;GO SEE IF STATIC ONLY
2968 013104 012737 013120 000706 MOV #LT43IT,SCOLP ;SET SCOPE ADDRESS
2969 013112 012737 026340 0C0616 MOV #MSLT43,EMADDR
2970 013120 004737 017002 LT43IT: JSR PC,INIT ;INIT, SELECT DRIVE+SLAVE
2971 013124 005001 CLR R1
2972 013126 005301 1$: DEC R1 ;DELAY
2973 013130 001376 BNE 1$
2974 013132 012777 000023 165374 MOV #23,@MR ;SET MM
2975 013140 012777 177770 165344 MOV #-10,@WC ;SET WC
2976 013146 012777 177760 165342 MOV #-20,@FC ;SET FC
2977 013154 012777 030040 165332 MOV #WDATA,@BA ;SET BA
2978 013162 012777 000061 165320 MOV #61,@C1 ;LOAD WRITE+GO
2979 013170 005000 CLR R0
2980 013172 032777 000200 165322 LT43C: BIT #200,@DS ;SEE IF DRY
2981 013200 001002 BNE LT43D ;IF SO: BR
2982 013202 005300 DEC R0
2983 013204 001372 BNE LT43C ;AWAIT DRY
2984 013206 032777 000200 165310 LT43D: BIT #200,@ER ;SEE IF LRC=1
2985 013214 001011 BNE LT43E ;IF SO: BR
2986 013216 012737 027577 000666 MOV #TMS26,ERADD ;SET ERROR CODE
2987 013224 012737 000001 000712 MOV #1,EXFL
2988 013232 004737 015220 JSR PC,LTGERO ;GO PRINT
2989 013236 000425 BR LT43X
2990 013240 017702 165270 LT43E: MOV @MR,R2
2991 013244 042702 000177 BIC #177,R2 ;MASK LRC
2992 013250 012701 157600 MOV #157600,R1 ;SET EXPT LRC
2993 013254 020102 CMP R1,R2 ;SEE IF EXPT = RCVD
2994 013256 001405 BEQ LT43F ;IF SO: BR
2995 013260 012737 023053 000666 MOV #MSG53,ERADD ;SET ERROR CODE
2996 013266 004737 016334 JSR PC,LTGER1 ;PRINT ERROR
2997 013272 017702 165226 LT43F: MOV @ER,R2 ;GET ERROR REGISTER
2998 013276 012701 000200 MOV #200,R1 ;SET EXPT ERROR BITS
2999 013302 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3000 013304 001402 BEQ LT43X ;IF NOT: BR
3001 013306 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
3002 013312 004737 016576 LT43X: JSR PC,ITER
3003 013316 004737 015666 JSR PC,DRVCLR
3004 013322 000137 002526 LT43XX: JMP TSCD2 ;RETURN TO SCHED
  
```



```

3005 ;LOGIC TEST 44: PE CORRECTABLE DATA (CORR)*****
3006
3007 013326 012737 002300 000772 LT44: MOV #2300,UDES ;SET UNIT DESCRIPTION = PE
3008 013334 004737 015022 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3009 013340 012737 026374 000616 MOV #MSLT44,EMADDR ;SET HEADER
3010 013346 012737 013354 000706 MOV #LT44IT,SCOLP ;SET SCOP
3011 013354 004737 017002 LT44IT: JSR PC,INIT ;GO INITIALIZE
3012 013360 012777 177600 165124 MOV #-200,@WC ;SET WC=200
3013 013366 012777 177400 165122 MOV #-400,@FC ;SET FC=400
3014 013374 012777 030040 165112 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
3015 013402 012777 000061 165100 MOV #61,@C1 ;LOAD WRITE AND GO
3016 013410 005000 CLR RO
3017 013412 005777 165100 LT44A: TST @FC ;SEE IF FC=0
3018 013416 001402 BEQ LT44A1 ;IF SO:BR
3019 013420 005300 DEC RO
3020 013422 001373 BNE LT44A ;AWAIT FC=0
3021 013424 012777 000021 165102 LT44A1: MOV #21,@MR ;SET MAINT MODE
3022 013432 005000 CLR RO
3023 013434 032777 000200 165060 LT44B: BIT #200,@DS ;SEE IF DRY
3024 013442 001002 BNE LT44C ;IF SO :BR
3025 013444 005300 DEC RO
3026 013446 001372 BNE LT44B ;AWAIT DRY
3027 013450 005777 165050 LT44C: TST @ER ;SEE IF CORR=1
3028 013454 100410 BMI LT44D ;IF SO: BR
3029 013456 012737 030000 000666 MOV #TMS35,ERADD ;ELSE SET ERROR CODE
3030 013464 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLAG
3031 013472 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
3032 013476 000240 LT44D: NOP
3033 013500 122777 000002 165022 LT44E: CMPB #2,@CC ;SEE IF DEAD TRACK BIT 1
3034 013506 001414 BEQ LT44F ;IF SO: BR
3035 013510 117702 165014 MOVB @CC,R2 ;ELSE SAVE RECVD
3036 013514 042702 177000 BIC #177000,R2 ;MASK OUT CRC
3037 013520 112701 000002 MOVB #2,R1 ;SAVE EXPT
3038 013524 012737 022462 000666 MOV #MSG42,ERADD ;SET ERROR CODE
3039 013532 004737 016334 JSR PC,LTGER1 ;GO PRINT ERROR
3040 013536 000410 BR LT44X
3041 013540 017702 164760 LT44F: MOV @ER,R2 ;GET ERROR REGISTER
3042 013544 012701 100000 MOV #100000,R1 ;SET EXPT ERROR BITS
3043 013550 020102 CMP R1,R2 ;SEE IF EXPT=RCVD
3044 013552 001402 BEQ LT44X ;IF SO: BR
3045 013554 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
3046 013560 004737 016576 LT44X: JSR PC,ITER ;GO SEE IF ITERATIONS
3047 013564 004737 015666 JSR PC,DRVCLR ;GO DO DRIVE CLEAR
3048 013570 000137 002526 LT44XX: JMP TSCD2 ;RETURN TO SCHED

```

```

3049
3050 ;LOGIC TEST 45: PE INCORRECTABLE DATA(INC)*****
3051
3052 013574 012737 002300 000772 LT45: MOV #2300,UDES ;SET UNIT DESCRIPTION = PE
3053 013602 004737 015022 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3054 013606 012737 026454 000616 MOV #MSLT45,EMADDR
3055 013614 012737 013622 000706 MOV #LT45IT,SCOLP
3056 013622 004737 017002 LT45IT: JSR PC,INIT ;INIT SELECT DRIVE SLAVE
3057 013626 012777 177600 164656 MOV #-200,@WC ;SET WC=200
3058 013634 012777 177400 164654 MOV #-400,@FC ;SET FC=400
3059 013642 012777 030040 164644 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
3060 013650 012777 000061 164632 MOV #61,@C1 ;LOAD WRITE+GO
3061 013656 005000 CLR RO
3062 013660 005777 164632 LT45E: TST @FC ;AWAIT FC=0
3063 013664 001402 BEQ LT45E1
3064 013666 005300 DEC RO
3065 013670 001373 BNE LT45E ;AWAIT FC=0
3066 013672 012777 000023 164634 LT45E1: MOV #23,@MR ;SET MAINT CODE
3067 013700 005000 CLR RO
3068 013702 032777 000200 164612 LT45A: BIT #200,@DS ;SEE IF DRY IS SET
3069 013710 001002 BNE LT45B ;IF SO: BR
3070 013712 005300 DEC RO
3071 013714 001372 BNE LT45A ;AWAIT DRY
3072 013716 032777 000100 164600 LT45B: BIT #100,@ER ;SEE IF INC=1
3073 013724 001010 BNE LT45D ;IF SO:BR
3074 013726 012737 027555 000666 MOV #TMS23,ERADD ;SET ERROR CODE
3075 013734 012737 000001 000712 MOV #1,EXFL
3076 013742 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
3077 013746 017702 164556 LT45D: MOV @CC,R2 ;GET CHECK CHAR
3078 013752 042702 177000 BIC #177000,R2 ;MASK CHECK CHAR
3079 013756 012701 000046 MOV #46,R1 ;SET EXPT CK
3080 013762 020102 CMP R1,R2 ;SEE IF EXPT = RCVD
3081 013764 001405 BEQ LT45F ;IF SO: BR
3082 013766 012737 023065 000666 MOV #MSG54,ERADD
3083 013774 004737 016334 JSR PC,LTGER1 ;ELSE GO PRINT ERROR
3084 014000 017702 164520 LT45F: MOV @ER,R2
3085 014004 042702 120600 BIC #120600,R2 ;MASK OPI ,NSG, CORR, AND PEF
3086 014010 012701 000100 MOV #100,R1 ;SET EXPT ERROR BITS
3087 014014 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3088 014016 001402 BEQ LT45X ;IF NOT: BR
3089 014020 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
3090 014024 004737 016576 LT45X: JSR PC,ITER
3091 014030 004737 015666 JSR PC,DRVCLR
3092 014034 000137 002526 LT45XX: JMP TSCD2 ;RETURN TO SCHED

```

```
3093
3094
3095
3096 014040 012737 00230C 000772 LT46: MOV #2300,UDES ;SET UNIT DESCRIPTION = PE
3097 014046 004737 015022 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3098 014052 012737 026536 000616 MOV #MSLT46,EMADDR ;SET HEADER
3099 014060 012737 014066 000706 MOV #LT46IT,SCOLP ;SET SCOPE ADDRESS
3100 014066 004737 017002 LT46IT: JSR PC,INIT ;INITIALIZE
3101 014072 012777 177770 164412 MOV #-10,@WC ;SET WC=10
3102 014100 012777 177760 164410 MOV #-20,@FC ;SET FC=20
3103 014106 012777 030040 164400 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
3104 014114 012777 000061 164366 MOV #61,@C1 ;LOAD WRITE+GO
3105 014122 005777 164370 LT46A: TST @FC
3106 014126 001375 BNE LT46A ;AWAIT FC=0
3107 014130 032777 000100 164376 1$: BIT #100,@MR ;WAIT FOR TAPE TO START WRITEING POSTAMBLE
3108 014136 001774 BEQ 1$ ;DELAY
3109 014140 032777 000100 164366 2$: BIT #100,@MR
3110 014146 001374 BNE 2$
3111 014150 012777 000027 164356 MOV #27,@MR ;SET MM CODE TO KILL PEF
3112 014156 005000 CLR R0 ;INIT TIMING LOOP
3113 014160 032777 000200 164334 LT46B: BIT #200,@DS ;SEE IF DRY SET
3114 014166 001002 BNE LT46C ;IF SO: BR
3115 014170 005300 DEC R0
3116 014172 001372 BNE LT46B ;AWAIT DRY
3117 014174 032777 000200 164322 LT46C: BIT #200,@ER ;SEE IF PEF SET
3118 014202 001011 BNE LT46D ;IF SO: BR
3119 014204 012737 027571 000666 MOV #TMS25,ERADD ;SET ERROR TAG
3120 014212 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLAG
3121 014220 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
3122 014224 000420 BR LT46X
3123 014226 017702 164272 LT46D: MOV @ER,R2 ;GET ERROR REGISTER
3124 014232 042702 120100 BIC #120100,R2 ;++B CLEAR CRC,OPI & INC BITS (MAY OR MAY NOT SET)
3125 014236 012701 000600 MOV #600,R1 ;++B SET EXPT ERROR BITS (NSG + PEF)
3126 014242 032777 000004 164266 BIT #4,@DT ;++B BRANCH IF TE16
3127 014250 001402 BEQ 1$ ;++B
3128 014252 042701 000400 BIC #400,R1 ;++B TU77 SHOULD NOT SET NSG BIT
3129 014256 020102 1$: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3130 014260 001402 BEQ LT46X ;IF NOT: BR
3131 014262 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
3132 014266 004737 016576 LT46X: JSR PC,ITER
3133 014272 004737 015666 JSR PC,DRVCLR
3134 014276 000137 002526 LT46XX: JMP TSCD2 ;RETURN TO SCHED
```

```

3135                                     ;LOGIC TEST 47: FRAME COUNT OVERFLOW(M8905-YB)*****
3136
3137 014302 012737 026572 000616 LT47:  MOV      #MSLT47,EMADDR  ;SET TEST HEADER
3138 014310 012737 014316 000706      MOV      #LT47IT,SCOLP  ;SET SCOPE ADDRESS
3139 014316 004737 016752      LT47IT: JSR      PC,INIT3  ;GO INIT
3140 014322 012777 177770 164162      MOV      #-10,@WC      ;SET WC = 10
3141 014330 012777 177760 164160      MOV      #-20,@FC      ;SET FC = 20
3142 014336 052777 001700 164176      BIS      #1700,@TC     ;SET TO NRZ, NORMAL, ODD
3143 014344 012777 030040 164142      MOV      #WDATA,@BA   ;SET BUS ADDRESS
3144 014352 012777 000013 164154      MOV      #13,@MR      ;SET WRAP 2
3145 014360 012777 000061 164122      MOV      #61,@C1      ;LOAD WRITE+GO
3146 014366 012700 040000      MOV      #40000,R0
3147 014372 005777 164144      LT47A:  TST      @TC      ;SEE IF ALPHA
3148 014376 100002      BPL      LT47B        ;IF SO: BR
3149 014400 005300      DEC      R0
3150 014402 001373      BNE      LT47A        ;AWAIT ALPHA
3151 014404 012700 000020      LT47B:  MOV      #20,R0  ;SET CLK CNT
3152 014410 052777 000040 164116      LT47C:  BIS      #40,@MR
3153 014416 042777 000040 164110      BIC      #40,@MR      ;CLOCK MR
3154 014424 005300      DEC      R0
3155 014426 001370      BNE      LT47C        ;IF NOT DONE ALL: BR
3156 014430 017702 164062      MOV      @FC,R2
3157 014434 005001      CLR      R1           ;SET TEST WORD
3158 014436 020102      CMP      R1,R2        ;SEE IF EXPT = RCVD
3159 014440 001410      BEQ      LT47X        ;IF SO: BR
3160 014442 012737 021735 000666      MOV      #MSG19,ERADD  ;SET ERROR CODE
3161 014450 012737 000001 000712      MOV      #1,EXFL      ;SET EXPT FLAG
3162 014456 004737 016334      JSR      PC,LTGER1    ;GO PRINT ERROR
3163 014462 004737 016576      LT47X:  JSR      PC,ITER ;GO SEE IF ITERATIONS
3164 014466 000137 002526      JMP      TSCD2        ;RETURN TO SCHEDULAR
3165

```

```

3166
3167
3168
3169 014472 012737 026642 000616 LT50: MOV #MSLT50,EMADDR ;SET ERROR POINTER FOR STANDARD
3170 014500 012737 014506 000706 MOV #LT50IT,SCOLP
3171 014506 004737 016752 LT50IT: JSR PC,INIT3 ;SET SLAVE = NRZ
3172 014512 042777 003400 164022 BIC #3400,@TC ;CLEAR DENSITY BITS
3173 014520 052777 002300 164014 BIS #2300,@TC ;SET DENSITY = PE
3174 014526 012777 177770 163756 MOV #-10,@WC ;SET WORD COUNT
3175 014534 012777 177760 163754 MOV #-20,@FC ;SET FRAME COUNT
3176 014542 012777 030040 163744 MOV #WDATA,@BA ;SET BUS ADDRESS
3177 014550 012777 000013 163756 MOV #13,@MR ;SET WRAP 2
3178 014556 012777 000061 163724 MOV #61,@C1 ;LOAD WRITE COMMAND
3179 014564 000240 NOP
3180 014566 000240 NOP
3181 014570 000240 NOP
3182 014572 012701 004000 2$: MOV #4000,R1 ;SET EXPECTED RESULT
3183 014576 017702 163722 3$: MOV @ER,R2 ;GET ERROR REGISTER
3184 014602 030102 BIT R1,R2 ;BRANCH IF NEF BIT SET
3185 014604 001006 BNE 1$
3186 014606 012737 000001 000712 MOV #1,EXFL ;SET EXPECTED FLAG
3187 014614 004737 015220 JSR PC,LTGERC ;PRINT ERROR
3188 014620 000404 BR LT50X
3189 014622 020102 1$: CMP R1,R2 ;BRANCH IF NO UNEXPECTED ERROR
3190 014624 001402 BEQ LT50X ;BITS WERE SET
3191 014626 004737 015206 JSR PC,LTGER3 ;PRINT ERROR MSG
3192 014632 004737 016576 LT50X: JSR PC,ITER ;+ITERATE TEST
3193 014636 004737 015666 JSR PC,DRVCLR ;RESET DRIVE
3194 014642 000137 002526 JMP TSCD2
3195
3196
  
```

```

3197
3198
3199
3200 014646 012737 026722 000616 LT51:  MOV #MSLT51,EMADDR ;+SET ERROR POINTER FOR STANDARD CONF.
3201 014654 012737 014662 000706      MOV #LT51IT,SCOLP ;SET SCOPE LOOP ADDRS.
3202 014662 004737 016766      LT51IT: JSR PC,INIT4 ;SET SLAVE = PE
3203 014666 042777 002300 163646      BIC #2300,@TC ;CLEAR DENSITY BITS
3204 014674 052777 001300 163640      BIS #1300,@TC ;SET DENSITY = NRZ
3205 014702 012777 177770 163602      MOV #-10,@WC ;SET WORD COUNT
3206 014710 012777 177760 163600      MOV #-20,@FC ;SET FRAME COUNT
3207 014716 012777 030040 163570      MOV #WDATA,@BA ;SET BUS ADDRESS
3208 014724 012777 000013 163602      MOV #13,@MR ;SET WRAP 2
3209 014732 012777 000061 163550      MOV #61,@C1 ;SET WRITE COMMAND AND GO
3210 014740 000240      NOP
3211 014742 000240      NOP
3212 014744 000240      NOP
3213 014746 012701 004000      MOV #4000,R1 ;SET EXPECTED RESULT
3214 014752 017702 163546      MOV @ER,R2 ;GET ERROR REGISTER
3215 014756 030102      BIT R1,R2 ;BRANCH IF NEF SET
3216 014760 001006      BNE 1$
3217 014762 012737 000001 000712      MOV #1,EXFL ;SET EXPECTED FLAG
3218 014770 004737 015220      JSR PC,LTGERO ;PRINT ERROR MSG
3219 014774 000404      BR LT51X
3220 014776 020102      1$:  CMP R1,R2 ;BRANCH IF NO UNEXPECTED
3221 015000 001402      BEQ LT51X ;ERROR BITS WERE SET
3222 015002 004737 015206      JSR PC,LTGER3
3223 015006 004737 016576      LT51X: JSR PC,ITER ;+ITERATE TEST
3224 015012 004737 015666      JSR PC,DRVCLR ;CLEAR DRIVE
3225 015016 000137 002526      JMP TSCD2 ;RETURN TO SCHEDULER
  
```

```
3226                                     ;STATIC TESTS ONLY SUBROUTINE*****  
3227  
3228 015022 005737 000734          STATIC: TST      STFLG      ;SEE IF SINGLE TEST ONLY  
3229 015026 001006                    BNE      1$      ;IF SO: BR  
3230 015030 005737 001006          TST      STATC   ;SEE IF STATIC ONLY  
3231 015034 001403                    BEQ      1$      ;IF NOT: BR  
3232 015036 005726                    TST      (SP)+   ;RESET STACK  
3233 015040 000137 002526          JMP      TSCD2   ;RETURN TO SCHEDULAR  
3234 015044 005037 001002          1$: CLR      RDRVF  
3235 015050 000207                    RTS      PC      ;RETURN TO TEST  
3236
```

```
3237  
3238  
3239 ;END OF RECORD FORCE SUBROUTINE*****  
3240 015052 017700 163456 EORPA: MOV @MR,R0 ;GET MAINT REG  
3241 015056 042700 000036 BIC #36,R0 ;CLEAR CURRENT OP CODE  
3242 015062 052700 000024 BIS #24,R0 ;SET EOR CLEAR OP CODE  
3243 015066 010077 163442 MOV R0,@MR ;DO EOR  
3244 015072 042777 000037 163434 BIC #37,@MR ;CLEAR EOR AND MM  
3245 015100 005000 CLR R0  
3246 015102 012701 000002 MOV #2,R1  
3247 015106 032777 000001 163374 EORP1: BIT #1,@C1 ;SEE IF GO GONE  
3248 015114 001430 BEQ EORP2 ;IF SO: BR  
3249 015116 005300 DEC R0  
3250 015120 001372 BNE EORP1 ;AWAIT GO RESET  
3251 015122 005301 DEC R1  
3252 015124 001370 BNE EORP1  
3253 015126 032777 020000 163434 BIT #20000,@SWR ;SEE IF ERROR PRINT INHIBIT  
3254 015134 001020 BNE EORP2 ;IF SO: BR  
3255 015136 005737 000614 TST HDRFL ;SEE IF DONE HEADER  
3256 015142 001004 BNE EORP1A ;IF SO: BR  
3257 015144 013704 000616 MOV EMADDR,R4  
3258 015150 004737 017724 JSR PC,TOUT ;PRINT HEADER  
3259 015154 012704 030453 EORP1A: MOV #WMSG31,R4  
3260 015160 004737 017724 JSR PC,TOUT ;PRINT EOR GO BIT ERROR  
3261 015164 032777 100000 163376 BIT #100000,@SWR ;SEE IF HALT ON ERROR  
3262 015172 001401 BEQ EORP2 ;IF NOT: BR  
3263 015174 000000 HALT  
3264 015176 000240 EORP2: NOP  
3265 015200 005037 000672 EORPX: CLR TEMP2 ;CLEAR FLAG  
3266 015204 000207 RTS PC ;RETURN  
3267
```



```

3268                                     ;LOGIC TEST ADDRESSING ERROR SUBROUTINE*****
3269
3270 015206 005037 000712          LTGER3: CLR      EXFL
3271 015212 012737 023024 000666  MOV      #MSG51,ERADD
3272 015220 012737 000001 000742  LTGER0: MOV      #1,ADDFL          ;SET NO ADDRESS FLAG
3273 015226 000240          LTGER:  NOP
3274 015230 005037 000662          CLR      PFLG          ;CLEAR PRINT FLAG
3275 015234 032777 020000 163326  BIT      #20000,@SWR      ;SEE IF SHOULD PRINT
3276 015242 001112          BNE     LTGX          ;IF NOT: BR
3277 015244 005737 000614          LTGA:  TST     HDRFL      ;SEE IF PRINTED HEADER
3278 015250 001004          BNE     LTGA1         ;IF SO: BR
3279 015252 013704 000616          MOV     EMADDR,R4
3280 015256 004737 017724          JSR    PC,TTOUT        ;PRINT TEST HEADER
3281 015262 012737 000001 000614  LTGA1: MOV     #1,HDRFL      ;SET HEADER FLAG
3282 015270 013704 000666          MOV     ERADD,R4
3283 015274 004737 017724          JSR    PC,TTOUT        ;PRINT CONDITION ERROR
3284 015300 005737 000742          TST    ADDFL
3285 015304 001003          BNE     LTGA2
3286 015306 010103          MOV     R1,R3
3287 015310 004737 020052          JSR    PC,OCTP        ;PRINT ADDRESS
3288 015314 005737 000712          LTGA2: TST     EXFL
3289 015320 001412          BEQ    LTGC          ;IF NO STATUS: BR
3290 015322 012704 021315          MOV     #MSG6,R4
3291 015326 022737 000001 000712  CMP     #1,EXFL        ;EXPT-NOT RCVD
3292 015334 001402          BEQ    LTGB
3293 015336 012704 021334          MOV     #MSG7,R4        ;RCVD-NOT EXPT
3294 015342 004737 017724          LTGB:  JSR    PC,TTOUT  ;PRINT STATUS
3295 015346 005237 000662          LTGC:  INC     PFLG
3296 015352 005737 000742          TST    ADDFL          ;SEE IF ADD TST
3297 015356 001430          BEQ    LTGD          ;IF SO: BR
3298 015360 005737 000740          TST    T24FL         ;SEE IF TEST 24
3299 015364 001423          BEQ    LTGCO         ;IF NOT: BR
3300 015366 012704 030440          MOV     #WMSG27,R4
3301 015372 004737 017724          JSR    PC,TTOUT        ;PRINT DATA TAG
3302 015376 012704 021635          MOV     #MSG12,R4
3303 015402 004737 017724          JSR    PC,TTOUT        ;PRINT EXPT TAG
3304 015406 012703 177777          MOV     #-1,R3
3305 015412 004737 020042          JSR    PC,OCTPE        ;PRINT EXPT
3306 015416 012704 021644          MOV     #MSG13,R4
3307 015422 004737 017724          JSR    PC,TTOUT        ;PRINT RCVD TAG
3308 015426 010103          MOV     R1,R3          ;GET RCVD
3309 015430 004737 020042          JSR    PC,OCTPE        ;PRINT RCVD
3310 015434 004737 015532          LTGCO: JSR    PC,REGP    ;PRINT REGISTERS
3311 015440 032777 004000 163122  LTGD:  BIT     #4000,@SWR
3312 015446 001010          BNE     LTGX
3313 015450 012704 021706          MOV     #MSG16,R4
3314 015454 004737 017724          JSR    PC,TTOUT
3315 015460 013703 000676          MOV     ITCNT,R3        ;PRINT ITERATION
3316 015464 004737 020052          JSR    PC,OCTP
3317 015470 005777 163074          LTGX:  TST     @SWR
3318 015474 100001          BPL    LTGXA          ;IF NOT STOP ON ERROR: BR
3319 015476 000000          HALT
3320 015500 005737 000662          LTGXA: TST     PFLG
3321 015504 001004          BNE     LTGXX          ;IF PRINTED: BR
3322 015506 032777 020000 163054  BIT     #20000,@SWR
3323 015514 001653          BEQ    LTGA

```

```
3324 015516 005037 000742      LTGXX: CLR      ADDFL      ;CLEAR ADDRESS FLAG
3325 015522 005037 000712      CLR      EXFL
3326 015526 000137 016546      JMP      SCOPE
3327
3328      ;SUBROUTINE TO PRINT MAJOR REGISTERS*****
3329
3330      REGP:  NOP
3331 015532 000240      MOV      #MSG46,R4
3332 015534 012704 022647      JSR      PC,TTOUT      ;PRINT REGISTER HEADER
3333 015544 017703 162740      MOV      @C1,R3
3334 015550 004737 020042      JSR      PC,OC1PE
3335 015554 017703 162732      MOV      @WC,R3
3336 015560 004737 020042      JSR      PC,OC1PE
3337 015564 017703 162724      MOV      @BA,R3
3338 015570 004737 020042      JSR      PC,OC1PE
3339 015574 017703 162716      MOV      @FC,R3
3340 015600 004737 020042      JSR      PC,OC1PE
3341 015604 017703 162710      MOV      @CS,R3
3342 015610 004737 020042      JSR      PC,OC1PE
3343 015614 017703 162702      MOV      @DS,R3
3344 015620 004737 020042      JSR      PC,OC1PE      ;PRINT REGISTERS
3345 015624 017703 162674      MOV      @ER,R3
3346 015630 004737 020042      JSR      PC,OC1PE
3347 015634 017703 162666      MOV      @AS,R3
3348 015640 004737 020042      JSR      PC,OC1PE
3349 015644 017703 162664      MOV      @MR,R3
3350 015650 004737 020042      JSR      PC,OC1PE
3351 015654 017703 162662      MOV      @TC,R3
3352 015660 004737 020042      JSR      PC,OC1PE
3353 015664 000207      RTS      PC
3354
3355
```

```

3356                                     ;DRIVE CLEAR SUBROUTINE*****
3357
3358 015666 000240          DRVCLR: NOP
3359 015670 012704 040000          MOV      #40000,R4
3360 015674 005304          DCD:    DEC      R4
3361 015676 001376          BNE     DCD          ;DELAY
3362 015700 005037 000662          CLR     PFLG
3363 015704 004737 016122          JSR     PC,ATTN      ;GO SEE OF ATTN SET
3364 015710 012777 000011 162572  MOV     #11,@C1      ;ISSUE DRIVE CLEAR
3365 015716 005000          CLR     R0
3366 015720 032777 000200 162574  DCA:    RIT     #200,@DS ;SEE IF DRY
3367 015726 001002          BNE     DCA0
3368 015730 005300          DEC     R0
3369 015732 001372          ENE     DCA          ;WAIT FOR DRY
3370 015734 032777 040000 162560  DCA0:  RIT     #40000,@DS ;SEE IF ERR RESET
3371 015742 001022          BNE     DCE          ;IF NOT: BR
3372 015744 005777 162554          TST     @ER          ;SEE IF ERROR REGISTER RESET
3373 015750 001017          BNE     DCE          ;IF NOT: BR
3374 015752 005777 162544          TST     @DS          ;SEE IF ATA RESET
3375 015756 100414          BMI    JCE          ;IF NOT: BR
3376 015760 012703 000001          MOV     #1,R3        ;SET TEST BIT
3377 015764 013704 000620          MOV     DRVN,R4      ;GET DRIVE NUMBER & BRANCH
3378 015770 001403          BEQ     DCC          ;IF DRIVE 0
3379 015772 006303          DCB:    ASL     R3        ;POSITION TEST BIT PER DRIVE NUMBER
3380 015774 005304          DEC     R4
3381 015776 001375          BNE     DCB          ;SEE IF DONE
3382 016000 030377 162522          DCC:    BIT     R3,@AS ;SEE IF ATTN IS RESET
3383 016004 001001          BNE     DCE          ;IF NOT: BR
3384 016006 000207          RTS     PC           ;RETURN
3385
3386 016010 000240          DCE:    NOP
3387 016012 032777 020000 162550  BIT     #20000,@SWR ;SEE IF ERROR PRINT INHIBIT
3388 016020 001017          BNE     DCEX         ;IF SO: BR
3389 016022 005737 000614          TST     HDRFL        ;SEE IF PRINT HEADER
3390 016026 001004          BNE     DCEA         ;IF NOT: BR
3391 016030 013704 000616          MOV     EMADDR,R4
3392 016034 004737 017724          JSR     PC,TTOUT      ;PRINT HEADER
3393 016040 012704 022753          DCEA:  MOV     #MSG47,R4
3394 016044 004737 017724          JSR     PC,TTOUT      ;PRINT DRIVE CLEAR ERROR
3395 016050 004737 015532          JSR     PC,REGP       ;PRINT REGISTERS
3396 016054 005237 000662          INC     PFLG         ;SET PRINTED FLAG
3397 016060 005777 162504          DCEX:  TST     @SWR    ;SEE IF HALT ON ERROR
3398 016064 100001          BPL     DCEXA        ;IF NOT: BR
3399 016066 000000          HALT
3400 016070 005737 000662          DCEXA: TST     PFLG      ;SEE IF HAVE PRINTED
3401 016074 001004          BNE     DCEXX        ;IF SO: BR
3402 016076 032777 020000 162464  BIT     #20000,@SWR ;BRANCH IF ERROR
3403 016104 001741          BEQ     DCE          ;PRINTOUT DESIRED
3404 016106 000240          DCEXX: NOP
3405 016110 012737 015666 000706  MOV     #DRVCLR,SCOLP ;SET SCOPE LOOP ADDRESS
3406 016116 000137 016546          JMP     SCOPE        ;GO DO SCOPE LOOP
  
```

```

3407                                     ;COMPOSITE ERROR CHECK SUBROUTINE*****
3408
3409 016122 000240          ATTN:  NOP
3410 016124 005777 162372      TST   @DS          ;SEE IF ATA SET
3411 016130 001004          BNE   ATTA          ;IF SO: BR
3412 016132 012737 022311 000674  MOV   #MSG32,TEMP3
3413 016140 000427          BR    ATTP          ;ELSE PRINT ERROR
3414 016142 032777 040000 162352  ATTA:  BIT   #40000,@DS  ;SEE IF COMPOSITE ERROR SET
3415 016150 001004          BNE   ATTB          ;IF SO: BR
3416 016152 012737 022273 000674  MOV   #MSG31,TEMP3
3417 016160 000417          BR    ATTP          ;ELSE PRINT ERROR
3418 016162 012703 000001          ATTB:  MOV   #1,R3      ;SET TEST BIT
3419 016166 012737 022327 000674  MOV   #MSG33,TEMP3
3420 016174 013704 000620          MOV   DRVN,R4      ;GET DRIVE NUMBER & BRANCH
3421 016200 001403          BEQ   ATTD          ;IF DRIVE 0
3422 016202 006303          ATTC:  ASL   R3      ;POSITION TEST BIT
3423 016204 005304          DEC   R4          ;SEE IF DONE
3424 016206 001375          BNE   ATTC          ;IF NOT: BR
3425 016210 030377 162312          ATTD:  BIT   R3,@AS  ;SEE IF ATTEN SUMMARY SET
3426 016214 001401          BEQ   ATTP          ;IF NOT: BR
3427 016216 000207          RTS   PC          ;ELSE RETURN
3428 016220 032777 020000 162342  ATTP:  BIT   #20000,@SWR ;SEE IF PRINT INHIBIT
3429 016226 001021          BNE   ATTX          ;IF SO: BR
3430 016230 005737 000614          TST   HDRFL        ;SEE IF DCNE HEADER
3431 016234 001004          BNE   ATTPA         ;IF SO: BR
3432 016236 013704 000616          MOV   EMADDR,R4
3433 016242 004737 017724          JSR   PC,TTOUT      ;PRINT HEADER
3434 016246 013704 000674          ATTPA: MOV   TEMP3,R4
3435 016252 004737 017724          JSR   PC,TTOUT      ;PRINT ERROR TYPE
3436 016256 004737 015532          JSR   PC,REGP       ;PRINT REGISTERS
3437 016262 005237 000662          INC   PFLG         ;SET PRINT FLAG
3438 016266 005237 000614          INC   HDRFL        ;SET HEADER FLAG
3439 016272 005777 162272          ATTX:  TST   @SWR    ;SEE IF HALT ON ERROR
3440 016276 100001          BPL   ATTXA         ;IF NOT: BR
3441 016300 000000          HALT
3442 016302 005737 000662          ATTXA: TST   PFLG     ;SEE IF DONE PRINT
3443 016306 001004          BNE   ATTX          ;IF SO: BR
3444 016310 032777 020000 162252  BIT   #20000,@SWR  ;BRANCH IF NO ERROR
3445 016316 001740          BEQ   ATTP          ;PRINTOUT DESIRED
3446 016320 005037 000662          ATTXX: CLR   PFLG    ;CLEAR PRINT FLAG
3447 016324 000207          RTS   PC          ;RETURN
  
```

```

3448                                     ;LOGIC TEST REGISTER BIT ERROR SUBROUTINE*****
3449
3450 016326 012737 000001 000732 LTGER2: MOV #1,PEXFL ;SET FLAG
3451 016334 000240 LTGER1: NOP
3452 016336 005037 000662 CLR PFLG ;CLEAR PRINT FLAG
3453 016342 032777 020000 162220 BIT #20000,@SWR ;BRANCH IF ERROR
3454 016350 001055 BNE LTG1X ;PRINTOUT DESIRED
3455 016352 005737 000614 LTG1A: TST HDRFL ;SEE IF PRINT HEADER
3456 016356 001004 BNE LTG1B ;IF NOT: BR
3457 016360 013704 000616 MOV EMADDR,R4
3458 016364 004737 017724 JSR PC,TTOUT ;PRINT HEADER
3459 016370 012737 000001 000614 LTG1B: MOV #1,HDRFL ;SET FLAG
3460 016376 013704 000666 MOV ERADD,R4
3461 016402 004737 017724 JSR PC,TTOUT ;PRINT ERROR CODE
3462 016406 005737 000732 TST PEXFL ;SEE IF PRINT EXPT-RCVD
3463 016412 001016 BNE LTG1T ;IF NOT: BR
3464 016414 012704 021635 MOV #MSG12,R4
3465 016420 004737 017724 JSR PC,TTOUT ;PRINT EXPT TAG
3466 016424 010103 MOV R1,R3
3467 016426 004737 020052 JSR PC,OCTP ;PRINT EXPT
3468 016432 012704 021644 MOV #MSG13,R4
3469 016436 004737 017724 JSR PC,TTOUT ;PRINT RCVD TAG
3470 016442 010203 MOV R2,R3
3471 016444 004737 020052 JSR PC,OCTP ;PRINT RCVD
3472 016450 032777 004000 162112 LTG1T: BIT #4000,@SWR
3473 016456 001010 BNE LTG1C
3474 016460 012704 021706 MOV #MSG16,R4
3475 016464 004737 017724 JSR PC,TTOUT
3476 016470 013703 000676 MOV ITCNT,R3
3477 016474 004737 020052 JSR PC,OCTP ;PRINT ITERATION
3478 016500 005237 000662 LTG1C: INC PFLG
3479 016504 000240 LTG1X: NOP
3480 016506 005777 162056 TST @SWR
3481 016512 100001 BPL LTG1X1 ;IF NOT STOP ON ERROR: BR
3482 016514 000000 HALT
3483 016516 005737 000662 LTG1X1: TST PFLG
3484 016522 001004 BNE LTG1XX ;IF HAVE PRINTED: BR
3485 016524 032777 020000 162036 BIT #20000,@SWR
3486 016532 001707 BEQ LTG1A
3487 016534 000240 LTG1XX: NOP
3488 016536 005037 000732 CLR PEXFL ;CLEAR EXPT-RCVD FLAG
3489 016542 000137 016546 JMP SCOPE ;GO TO SCOPE
3490
3491                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
3492
3493
3494 016546 000240 SCOPE: NOP
3495 016550 032777 040000 162012 BIT #40000,@SWR ;SEE IF LOOP ON ERROR
3496 016556 001001 BNE 1$ ;IF SO: BR
3497 016560 000207 RTS PC ;ELSE EXIT
3498 016562 000240 1$: NOP
3499 016564 005726 TST (SP)+ ;RESET STACK
3500 016566 000240 NOP
3501 016570 000240 NOP
3502 016572 000177 162110 JMP @SCOLP ;LOOP ON ERROR
3503

```

```
3504 ;TEST ITERATION SUBROUTINE*****
3505
3506 016576 032777 004000 161764 ITER: BIT #4000,@SWR ;SEE IF ITERATIONS
3507 016604 001403 BEQ 2$ ;IF SO: BR
3508 016606 005037 000676 1$: CLR ITCNT ;CLEAR ITERATION COUNTER
3509 016612 000207 RTS PC ;ELSE EXIT
3510 016614 005737 001014 2$: TST PCNTR ;NO SUBTEST ITERATIONS ON FIRST PASS
3511 016620 001772 BEQ 1$
3512 016622 005237 000676 INC ITCNT ;BUMP COUNTER
3513 016626 023737 000676 000602 CMP ITCNT,ITAMT ;SEE IF DONE ALL
3514 016634 001764 BEQ 1$ ;IF SO: BR
3515 016636 005726 TST (SP)+ ;RESET STACK
3516 016640 017700 162044 MOV @ITRLP,R0 ;SET ITERATION POINTER
3517 016644 000110 JMP (R0) ;GO ITERATE
3518
3519 ;MANUAL INTERVENTION INHIBIT*****
3520
3521 016646 000240 INMT: NOP
3522 016650 012704 022477 MOV #MSG43,R4
3523 016654 004737 017724 JSR PC,TTOUT ;GO PRINT INHIB MSG
3524 016660 000000 HALT
3525 016662 000137 002526 JMP TSCD2 ;RETURN TO SCHED
3526
3527
3528 ;NON-STANDARD MODE TEST HANDLER
3529
3530 016666 010046 NOST: MOV R0,-(SP) ;+SAVE R0
3531 016670 012700 000240 MOV #240,R0 ;+SET UP INDEX
3532 016674 013760 001326 001056 MOV TADX,TSTTBL(R0) ;+
3533 016702 005720 TST (R0)+
3534 016704 012737 000047 001330 MOV #47,TLAST ;+SET END OF TEST
3535 016712 013760 001330 001056 MOV TLAST,TSTTBL(R0) ;+SET LAST TEST NUMBER
3536 016720 012600 MOV (SP)+,R0 ;+RESTORE R0
3537 016722 000207 RTS PC ;+RETURN
3538
```

```

3539
3540                                     ;INITIALIZE SUBROUTINE*****
3541
3542 016724 000240      INIT1:  NOP
3543 016726 012777 000040 161564      MOV    #40,ACS      ;INIT
3544 016734 013777 000620 161556      MOV    DRVN,ACS    ;SELECT DRIVE
3545 016742 013777 000660 161572      MOV    SLVN,ATC    ;SELECT SLAVE
3546 016750 000207      RTS      PC      ;RETURN
3547
3548                                     ;ROUTINES TO INITIALIZE SLAVE. THESE ROUTINES PLACE THE SLAVE
3549                                     ;IN PROPER STATUS FOR THE CALLING TEST. INIT3 PLACES THE SLAVE IN
3550                                     ;NRZ MODE AND OFF BOT; INIT4 PLACES THE SLAVE IN PE MODE AND OFF
3551                                     ;BOT. IF THE SLAVE IS IN THE PROPER STATUS ON ENTRY NO ACTION IS TAKEN.
3552
3553                                     ;SET SLAVE IN NRZ OFF BOT
3554 016752 013746 000772      INIT3:  MOV    UDES,-(SP)    ;SAVE TEST'S UNIT DESCRIPTION
3555 016756 012737 001400 000772      MOV    #1400,UDES  ;SET UNIT DESCRIPTION = NRZ
3556 016764 000410      BR      INIT5      ;GO TO INIT5 ROUTINE
3557
3558                                     ;SET SLAVE IN PE OFF BOT
3559 016766 013746 000772      INIT4:  MOV    UDES,-(SP)    ;SAVE TEST'S UNIT DESCRIPTION
3560 016772 012737 002000 000772      MOV    #2000,UDES  ;SET UNIT DESCRIPTION = PE
3561 017000 000402      BR      INIT5      ;GO DO IT
3562
3563                                     ;THIS ROUTINE IS ENTERED AT INIT WHEN THE CALLER HAS SETUP UDES.
3564                                     ;IT IS ENTERED AT INIT5 WHEN EITHER INIT3 OR INIT4 HAS SET UP UDES.
3565 017002 013746 000772      INIT:   MOV    UDES,-(SP)    ;SAVE TEST'S UNIT DESCRIPTION
3566 017006 012777 000040 161504      INIT5:  MOV    #40,ACS      ;INIT CONTROLLER
3567 017014 013777 000620 161476      MOV    DRVN,ACS    ;SELECT TMO3 DRIVE
3568 017022 013777 000660 161512      MOV    SLVN,ATC    ;SELECT TE16 SLAVE
3569 017030 013746 000772      MOV    UDES,-(SP)    ;GET SLAVE DESCRIPTION
3570 017034 042716 174377      BIC    #174377,(SP)  ;CLEAR ALL BUT DENSITY SELECT BITS
3571 017040 022726 001400      CMP    #1400,(SP)+  ;BRANCH IF REQUESTING PE MODE
3572 017044 001005      BNE    1$
3573 017046 032777 000040 161446      BIT    #40,ADS      ;BRANCH IF SLAVE IS IN NRZ MODE
3574 017054 001420      BEQ    4$           ;(PES = 0)
3575 017056 000404      BR      2$
3576 017060 032777 000040 161434 1$:  BIT    #40,ADS      ;BRANCH IF SLAVE IS IN PE MODE
3577 017066 001013      BNE    4$
3578 017070 012777 000007 161412 2$:  MOV    #7,AC1      ;REWIND SLAVE
3579 017076 032777 000200 161416 20$: BIT    #200,ADS     ;WAIT FOR READY
3580 017104 001774      BEQ    20$
3581 017106 032777 020000 161406 3$:  BIT    #20000,ADS   ;WAIT UNTIL PIP CLEARS
3582 017114 001374      BNE    3$
3583 017116 053777 000772 161416 4$:  BIS    UDES,ATC    ;LOAD SLAVE DESCRIPTION
3584 017124 032777 000002 161370      BIT    #2,ADS      ;BRANCH IF NOT AT BOT
3585 017132 001407      BEQ    6$
3586 017134 012777 000025 161346      MOV    #25,AC1     ;ERASE TO GET OFF BOT
3587 017142 032777 000200 161352 5$:  BIT    #200,ADS     ;WAIT FOR READY
3588 017150 001774      BEQ    5$
3589 017152 032777 000020 161342 6$:  BIT    #20,ADS     ;WAIT FOR SETTLEDOWN TO CLEAR
3590 017160 001374      BNE    6$
3591 017162 012777 000011 161320      MOV    #11,AC1     ;RESET DRIVE
3592 017170 012637 000772      MOV    (SP)+,UDES  ;RESTORE UNIT DESCRIPTION
3593 017174 000207      RTS      PC      ;RETURN
3594

```

```
3595
3596
3597
3598
3599 017176 000240
3600 017200 004737 017724
3601 017204 012704 027002
3602 017210 004737 017724
3603 017214 012705 000674
3604 017220 012701 000001
3605 017224 012702 177777
3606 017230 012703 000000
3607 017234 004737 017402
3608 017240 000240
3609 017242 000207
3610
3611
3612
3613 017244 000240
3614 017246 013716 000664
3615 017252 000002
3616
3617
3618
3619 017254 017746 161314
3620 017260 042716 000200
3621 017264 122716 000003
3622 017270 001010
3623 017272 005737 001420
3624 017276 001005
3625 017300 005077 161262
3626 017304 000005
3627 017306 000137 000200
3628 017312 122716 000001
3629 017316 001017
3630 017320 022737 000176 000570
3631 017326 001016
3632 017330 012737 177570 000570
3633 017336 004737 020576
3634 017342 012704 023474
3635 017346 004737 017724
3636 017352 004737 020620
3637 017356 122716 000007
3638 017362 001005
3639 017364 012737 000176 000570
3640 017372 004737 020500
3641 017376 005726
3642 017400 000002
3643
```

```

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

;MAG TAPE INTERRUPT HANDLER*****
MTINT: NOP
      MOV  RTRN,(SP)    ;SET RETURN FROM INTERUPT ADDRESS
      RTI                ;RETURN

;TTY INTERRUPT HANDLER*****
TTINT: MOV  @TKB,-(SP)   ;GET CHARACTER
      BIC  #200,(SP)    ;CLEAR PARITY BIT
      CMPB #3,(SP)      ;BRANCH IF NOT CONTROL C
      BNE  1$
      TST  CHNFLG       ;INHIBIT ^C IF IN CHAIN MODE
      BNE  1$
      CLR  @PSW         ;CLEAR PSW
      RESET
      JMP  @#200        ;RESTART
1$:   CMPB  #1,(SP)     ;BRANCH IF NOT ^A
      BNE  2$
      CMP  #SWREG,SWR   ;BRANCH IF USING HARWARE SWR
      BNE  3$
      MOV  #177570,SWR  ;INVOKE HARDWARE SWR
      JSR  PC,.SAVE     ;SAVE REGISTERS ON THE STACK
      MOV  #MSG63,R4    ;TYPE 'HARDWARE SWR IN USE'
      JSR  PC,TTOUT
      JSR  PC,.RESTORE
2$:   CMPB  #7,(SP)     ;BRANCH IF NOT ^G
      BNE  4$
3$:   MOV  #SWREG,SWR   ;INVOKE SOFTWARE SWR
      JSR  PC,GTSWR    ;GET SWITCHES
4$:   TST  (SP)+        ;POP CHARACTER OFF STACK
      RTI                ;RETURN
```



```

3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661 017402 010146
3662 017404 011601
3663 017406 005037 000670
3664 017412 005000
3665 017414 004737 017662
3666 017420 122737 000003 000612
3667 017426 001003
3668 017430 000005
3669 017432 000137 000200
3670 017436 122737 000015 000612
3671 017444 001004
3672 017446 005737 000670
3673 017452 001471
3674 017454 000457
3675 017456 122737 000025 000612
3676 017464 001005
3677 017466 012704 023422
3678 017472 004737 017724
3679 017476 000742
3680 017500 122737 000177 000612
3681 017506 001012
3682 017510 000241
3683 017512 006000
3684 017514 006200
3685 017516 006200
3686 017520 012704 023424
3687 017524 004737 017724
3688 017530 005201
3689 017532 000730
3690 017534 122737 000060 000612
3691 017542 101402
3692 017544 000137 017642
3693 017550 122737 000070 000612
3694 017556 101002
3695 017560 000137 017642
3696 017564 005237 000670
3697 017570 006300
3698 017572 006300
3699 017574 006300

```

```

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

```

```

TTR: MOV R1, -(SP) ;SAVE CHARACTER COUNT
10$: MOV (SP), R1 ;RESTORE CHARACTER COUNT (FOR ^U)
CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
CLR R0
1$: JSR PC, TTIN ;GO READ CHARACTER
CMPB #3, TIB ;BRANCH IF NOT ^C
BNE 11$
RESET ;RESET
JMP @#200 ;RESTART PROGRAM
11$: CMPB #15, TIB ;SEE IF CR
BNE 2$ ;IF NOT: BR
TST TEMP1 ;SEE IF FIRST CHARACTER
BEQ 9$ ;IF SO: BR
BR 6$
2$: CMPB #25, TIB ;BRANCH IF NOT CONTROL U
BNE 21$
MOV #MSG59, R4 ;TYPE <CR><LF>
JSR PC, TTOUT
BR 10$
21$: CMPB #177, TIB ;BRANCH IF NOT 'RUBOUT'
BNE 3$
CLC
ROR R0 ;REMOVE LAST CHAR
ASR R0
ASR R0
MOV #MSG60, R4 ;TYPE '\ '
JSR PC, TTOUT
INC R1 ;DECREMENT CHARS RECEIVED COUNT
BR 1$
3$: CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0
BLOS 4$ ;IF NOT: BR
JMP TIB ;ELSE GO TO ERROR
4$: CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7
BHI 5$ ;IF NOT: BR
JMP TIB ;ELSE GO TO ERROR
5$: INC TEMP1 ;SET FIRST CHARACTER FLAG
ASL R0
ASL R0 ;SHIFT 3 LEFT
ASL R0

```

3700	017576	042737	177770	000612	BIC	#177770,T1B	;STRIP ASCII
3701	017604	053700	000612		BIS	T1B,R0	;LOAD CHARACTER
3702	017610	005301			DEC	R1	;SEE IF DONE
3703	017612	001300			BNE	1\$;IF NOT: BR
3704	017614	020002		6\$:	CMP	R0,R2	;SEE IF EXCEEDED MAXIMUM LIMIT
3705	017616	101402			BLOS	7\$;IF NOT: BR
3706	017620	000137	017642		JMP	T1NER	;ELSE GO TO ERROR
3707	017624	020300		7\$:	CMP	R3,R0	;SEE IF BELOW MINIMUM LIMIT
3708	017626	101402			BLOS	8\$;IF NOT: BR
3709	017630	000137	017642		JMP	T1NER	;ELSE GO TO ERROR
3710	017634	010015		8\$:	MOV	R0,(R5)	;LOAD VALUE
3711	017636	005726		9\$:	TST	(SP)+	;POP CHAR COUNT OFF STACK
3712	017640	000207			RTS	PC	;EXIT
3713							

```

3714
3715 ;TTY ENTRY ERROR SUBROUTINE*****
3716
3717 017642 012704 022437 T1NER: MOV #MSG40,R4
3718 017646 004737 017724 JSR PC,TTOUT ;PRINT?
3719 017652 005726 TST (SP)+ ;POP CHAR COUNT OFF STACK
3720 017654 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
3721 017660 000207 RTS PC ;REDO VALUE ENTRY
3722
  
```

```

3723 ;TTY READ SUBROUTINE*****
3724
3725 017662 005277 160704 TTIN: INC @TKS
3726 017666 105777 160700 1$: TSTB @TKS
3727 017672 100375 BPL 1$
3728 017674 017737 160674 000612 MOV @TKB,TIB
3729 017702 042737 000200 000612 BIC #200,TIB ;STRIP PARITY BIT
3730 017710 013737 000612 000610 MOV TIB,TOB ;MOVE CHAR TO TTY OUTPUT BFR
3731 017716 004737 020024 JSR PC,TOG ;AND ECHO IT
3732 017722 000207 RTS PC
3733
  
```

```

3734 ;TTY OUTPUT SUBROUTINE*****
3735
3736 017724 112437 000610 TTOUT: MOVB (R4)+,TOB
3737 017730 122737 000043 000610 CMPB #43,TOB
3738 017736 001440 BEQ TEX
3739 017740 122737 000045 000610 CMPB #45,TOB
3740 017746 001403 BEQ 1$
3741 017750 004737 020024 JSR PC,TOG
3742 017754 000763 BR TTOUT
3743 017756 112737 000015 000610 1$: MOVB #15,TOB
3744 017764 004737 020024 JSR PC,TOG
3745 017770 012703 000004 MOV #4,R3
3746 017774 005037 000610 2$: CLR TOB
3747 020000 004737 020024 JSR PC,TOG
3748 020004 005303 DEC R3
3749 020006 001372 BNE 2$ ;DO FILLERS
3750 020010 112737 000012 000610 MOVB #12,TOB
3751 020016 004737 020024 JSR PC,TOG
3752 020022 000740 BR TTOUT
3753 020024 105777 160546 TOG: TSTB @TPS
3754 020030 100375 BPL TOG
3755 020032 113777 000610 160540 MOVB TOB,@TPB
3756 020040 000207 TEX: RTS PC
3757
  
```

```

3758 ;OCTAL OUTPUT SUBROUTINE*****
3759
3760
3761 020042 012737 000001 020272 OCTPE: MOV #1,OFL
3762 020050 000402 BR OCTPE1
3763 020052 005037 020272 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
3764 020056 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
3765 020060 001006 BNE OCTP0 ;IF NOT ZERO: BR
3766 020062 005737 020272 TST OFL ;SEE IF PRINT ALL 0
3767 020066 001003 BNE OCTP0 ;IF SO: BR
3768 020070 004737 020252 JSR PC,OCTPG1 ;ELSE PRINT ZERO
3769 020074 000447 BR OCTP3 ;SPACE AND EXIT
  
```

```

3770 020076 032704 100000      OCTP0:  BIT      #100000,R4      ;SEE IF MSD = 1
3771 020102 001405              BEQ      OCTP1              ;IF NOT: BR
3772 020104 012704 000001      MOV      #1,R4
3773 020110 004737 020230      JSR      PC,OCTPG          ;PRINT 1
3774 020114 000403              BR       OCTP2
3775 020116 005004      OCTP1:  CLR      R4
3776 020120 004737 020230      JSR      PC,OCTPG          ;PRINT 0
3777 020124 010304      OCTP2:  MOV      R3,R4
3778 020126 006004              ROR      R4
3779 020130 006004              ROR      R4
3780 020132 006004              ROR      R4              ;POSITION DIGIT
3781 020134 006004              ROR      R4
3782 020136 000304              SWAB     R4
3783 020140 004737 020230      JSR      PC,OCTPG          ;PRINT DIGIT 2
3784 020144 010304              MOV      R3,R4
3785 020146 006004              ROR      R4
3786 020150 000304              SWAB     R4
3787 020152 004737 020230      JSR      PC,OCTPG          ;PRINT DIGIT 3
3788 020156 010304              MOV      R3,R4
3789 020160 006104              ROL      R4
3790 020162 006104              ROL      R4
3791 020164 000304              SWAB     R4
3792 020166 004737 020230      JSR      PC,CCTPG          ;PRINT DIGIT 4
3793 020172 010304              MOV      R3,R4
3794 020174 006004              ROR      R4
3795 020176 006004              ROR      R4
3796 020200 006004              ROR      R4
3797 020202 004737 020230      JSR      PC,CCTPG          ;PRINT DIGIT 5
3798 020206 010304              MOV      R3,R4
3799 020210 004737 020230      JSR      PC,OCTPG          ;PRINT DIGIT 5
3800 020214 012737 000240 000610  OCTP3:  MOV      #240,TOB
3801 020222 004737 020024      JSR      PC,TOG
3802 020226 000207      RTS      PC              ;EXIT
3803
3804 020230 042704 177770      OCTPG:  BIC      #177770,R4
3805 020234 001004              BNE     OCTPG0
3806 020236 005737 020272      TST     OFL
3807 020242 001001              BNE     OCTPG0
3808 020244 000207      RTS      PC
3809
3810 020246 005237 020272      OCTPG0:  INC     OFL
3811 020252 052704 000260      OCTPG1:  BIS     #260,R4
3812 020256 010437 000610      MOV     R4,TOB
3813 020262 004737 020024      JSR     PC,TOG
3814 020266 010304              MOV     R3,R4
3815 020270 000207      RTS     PC
3816 020272 000000      OFL:    0              ;FIRST CHAR FLAG
3817
  
```

```

3818
3819
3820 ;DATA CHARACTER OUTPUT SUBROUTINE*****
3821 020274 012704 000010 DOUT: MOV #10,R4 ;SET NUMBER TO PRINT
3822 020300 110337 000610 MOVB R3,TOB
3823 020304 105777 160266 1$: TSTB @TPS
3824 020310 100375 BPL 1$
3825 020312 132737 000200 000610 BITB #200,TOB
3826 020320 001404 BEQ 2$
3827 020322 012777 000061 160250 MOV #061,@TPB
3828 020330 000403 BR 3$
3829 020332 012777 000060 160240 2$: MOV #060,@TPB
3830 020340 006337 000610 3$: ASL TOB
3831 020344 005304 DEC R4
3832 020346 001356 BNE 1$
3833 020350 000207 RTS PC
3834
3835 020352 013703 000674 DOUTD: MOV TEMP3,R3
3836 020356 000303 SWAB R3
3837 020360 004737 020274 JSR PC,DOUT
3838 020364 013703 000674 MOV TEMP3,R3
3839 020370 004737 020274 JSR PC,DOUT
3840 020374 000207 RTS PC
3841
3842 ;TE16 SERIAL NUMBER PRINT SUBROUTINE*****
3843
3844 020376 010304 SNPT: MOV R3,R4
3845 020400 000304 SWAB R4
3846 020402 006004 ROR R4
3847 020404 006004 ROR R4
3848 020406 006004 ROR R4
3849 020410 006004 ROR R4 ;GET FIRST DIGIT
3850 020412 004737 020454 JSR PC,SNPG ;PRINT
3851 020416 010304 MOV R3,R4
3852 020420 000304 SWAB R4 ;GET SECOND DIGIT
3853 020422 004737 020454 JSR PC,SNPG ;PRINT
3854 020426 010304 MOV R3,R4
3855 020430 006004 ROR R4
3856 020432 006004 ROR R4
3857 020434 006004 ROR R4
3858 020436 006004 ROR R4
3859 020440 004737 020454 JSR PC,SNPG ;PRINT THIRD DIGIT
3860 020444 010304 MOV R3,R4
3861 020446 004737 020454 JSR PC,SNPG ;PRINT FOURTH DIGIT
3862 020452 000207 RTS PC ;EXIT
3863 020454 012737 000260 000610 SNPG: MOV #260,TOB ;SET BASE = 0
3864 020462 042704 177760 BIC #177760,R4 ;MASK DIGIT
3865 020466 050437 000610 BIS R4,TOB ;SET ASCII
3866 020472 004737 020024 JSR PC,TOG ;TYPE DIGIT
3867 020476 000207 RTS PC ;RETURN

```

```

3868
3869 ;ROUTINE TO LOAD CONTENTS OF SOFTWARE SWITCH REGISTER.
3870 ;IF A CONTROL G (^G) IS TYPED THE SOFTWARE SWITCH REGISTER IS LOADED
3871 020500 022737 000176 000570 GTSWR: CMP #SWREG,SWR ;BRANCH IF SOFTWARE SWR
3872 020506 001032 BNE 1$ ;NOT INVOKED
3873 020510 004737 020576 JSR PC,SAVE ;SAVE REGISTERS ON THE STACK
3874 020514 012704 027327 MOV #SMSWR,R4 ;TYPE 'SWR = '
3875 020520 004737 017724 JSR PC,TTOUT
3876 020524 017703 160040 MOV @SWR,R3 ;GET CURRENT VALUE
3877 020530 004737 020042 JSR PC,OCTPE ;AND TYPE IT
3878 020534 012704 027337 MOV #MNEW,R4 ;ASK FOR NEW VALUE
3879 020540 004737 017724 JSR PC,TTOUT
3880 020544 013705 000570 MOV SWR,R5 ;NEW VALUE WILL BE RETURNED IN (R5)
3881 020550 012701 000007 MOV #7,R1 ;LIMIT TO 7 CHARACTERS
3882 020554 012702 177777 MOV #177777,R2 ;LIMIT RESPONSE TO BETWEEN
3883 020560 012703 000000 MOV #0,R3 ;0 AND 177777
3884 020564 004737 017402 JSR PC,TTR ;GET RESPONSE
3885 020570 004737 020620 JSR PC,.RESTORE ;RESTORE REGISTERS
3886 020574 000207 1$: RTS PC ;RETURN TO CALLER
3887 ;:ROUTINE TO SAVE REGISTERS ON THE STACK
3888 020576 010546 .SAVE: MOV %5,-(SP) ;;R5 IS SAVED AT 12(SP)
3889 020600 010446 MOV %4,-(SP) ;;R4 IS SAVED AT 10(SP)
3890 020602 010346 MOV %3,-(SP) ;;R3 IS SAVED AT 6(SP)
3891 020604 010246 MOV %2,-(SP) ;;R2 IS SAVED AT 4(SP)
3892 020606 010146 MOV %1,-(SP) ;;R1 IS SAVED AT 2(SP)
3893 020610 010046 MOV %0,-(SP) ;;R0 IS SAVED AT (SP)
3894 020612 016646 000014 MOV 14(SP),-(SP) ;;PUSH RETURN PC ON THE STACK
3895 020616 000207 RTS PC ;;RETURN TO CALLER
3896
3897 ;:ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
3898 020620 012666 000014 .RESTORE:MOV (SP)+,14(SP) ;;STORE RETURN PC ON STACK
3899 020624 012600 MOV (SP)+,%0
3900 020626 012601 MOV (SP)+,%1
3901 020630 012602 MOV (SP)+,%2
3902 020632 012603 MOV (SP)+,%3
3903 020634 012604 MOV (SP)+,%4
3904 020636 012605 MOV (SP)+,%5
3905 020640 000207 RTS PC ;;RETURN
3906
3907
3908 ;MESSAGE TABLE*****
3909
3910 020642 022445 046524 031460 MSG1: .ASCII '%TM03-TE16/TU77 CONTROL LOGIC TEST- PART I (CZTEACO)';++B
3911 020650 052055 030505 027466
3912 020656 052524 033467 041440
3913 020664 047117 051124 046117
3914 020672 046040 043517 041511
3915 020700 052040 051505 026524
3916 020706 050040 051101 020124
3917 020714 020111 041450 052132
3918 020722 040505 030103 051
3919 020727 045 025052 040452 .ASCII /%***ASSURE TAPE IS AT BOT***/
3920 020734 051523 051125 020105
3921 020742 040524 042520 044440
3922 020750 020123 052101 041040
3923 020756 052117 025052 052

```

CZTEACO TM03-TE16/TU77 CTL 1
CZTEAC.P11 14-MAY-79 09:02

MACY11 30A(1052) 14-MAY-79 09:06 ^{L 8} PAGE 102

SEQ 0102

3924	020763	045	054524	042520		.ASCII /%TYPE <CR> TO TERMINATE RESPONSE & ^C TO RESTART%/
3925	020770	036040	051103	020076		
3926	020776	047524	052040	051105		
3927	021004	044515	040516	042524		
3928	021012	051040	051505	047520		
3929	021020	051516	020105	020046		
3930	021026	041536	052040	020117		
3931	021034	042522	052123	051101		
3932	021042	022524	043			
3933	021045	045	051104	053111	MSG2:	.ASCII /%DRIVE NUMBER OR (CR) WHEN DONE #/
3934	021052	020105	052516	041115		
3935	021060	051105	047440	020122		
3936	021066	041450	024522	053440		
3937	021074	042510	020116	047504		
3938	021102	042516	021440			
3939	021106	022445	047506	020122	MSG2A:	.ASCII /%%FOR DRIVE ADDRESS TEST;/
3940	021114	051104	053111	020105		
3941	021122	042101	051104	051505		
3942	021130	020123	042524	052123		
3943	021136	073				
3944	021137	045	042440	052116		.ASCII /% ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
3945	021144	051105	042440	050130		
3946	021152	020124	051104	053111		
3947	021160	020105	052516	041115		
3948	021166	051105	020054	046101		
3949	021174	020114	052117	042510		
3950	021202	051522	051440	047510		
3951	021210	046125	020104	042502		
3952	021216	047040	047117	042455		
3953	021224	044530	052123	047101		
3954	021232	027124	043			
3955	021235	045	047516	026516	MSG3:	.ASCII /%NON-EXIST DRIVE #/
3956	021242	054105	051511	020124		
3957	021250	051104	053111	020105		
3958	021256	043				
3959	021257	045	044122	042040	MSG4:	.ASCII /%RH DETECTED #/
3960	021264	052105	041505	042524		
3961	021272	020104	043			
3962	021275	045	046524	031460	MSG5:	.ASCII /%TM03 DETECTED #/
3963	021302	042040	052105	041505		
3964	021310	042524	020104	043		
3965	021315	105	050130	026524	MSG6:	.ASCII /EXPT-NOT RECVD#/
3966	021322	047516	020124	042522		
3967	021330	053103	021504			
3968	021334	041522	042126	047055	MSG7:	.ASCII /RCVD-NOT EXPT#/
3969	021342	052117	042440	050130		
3970	021350	021524				
3971	021352	051445	040514	042526	MSG8:	.ASCII /%SLAVE NUMBER OR (CR) WHEN DONE #/
3972	021360	047040	046525	042502		
3973	021366	020122	051117	024040		
3974	021374	051103	020051	044127		
3975	021402	047105	042040	047117		
3976	021410	020105	043			
3977	021413	045	043045	051117	MSG8A:	.ASCII /%%FOR SLAVE ADDRESS TEST;/
3978	021420	051440	040514	042526		
3979	021426	040440	042104	042522		

3980	021434	051523	052040	051505	
3981	021442	035524			
3982	021444	020045	047105	042524	.ASCII /% ENTER EXPT SLAVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
3983	021452	020122	054105	052120	
3984	021460	051440	040514	042526	
3985	021466	047040	046525	042502	
3986	021474	026122	040440	046114	
3987	021502	047440	044124	051105	
3988	021510	020123	044123	052517	
3989	021516	042114	041040	020105	
3990	021524	047516	026516	054105	
3991	021532	051511	040524	052116	
3992	021540	021456			
3993	021542	047045	047117	042455	MSG9: .ASCII /%NON-EXIST SLAVE #/
3994	021550	044530	052123	051440	
3995	021556	040514	042526	021440	
3996	021564	051045	040505	020104	MSG10: .ASCII /%READ CONT BUS PAR #/
3997	021572	047503	052116	041040	
3998	021600	051525	050040	051101	
3999	021606	021440			
4000	021610	053445	044522	042524	MSG11: .ASCII /%WRITE CONT BUS PAR #/
4001	021616	041440	047117	020124	
4002	021624	052502	020123	040520	
4003	021632	020122	043		
4004	021635	040	054105	052120	MSG12: .ASCII / EXPT #/
4005	021642	021440			
4006	021644	051040	053103	020104	MSG13: .ASCII / RCVD #/
4007	021652	043			
4008	021653	045	051115	041040	MSG14: .ASCII /%MR BITS 4-0#/
4009	021660	052111	020123	026464	
4010	021666	021460			
4011	021670	046445	020122	044502	MSG15: .ASCII /%MR BITS 15-7#/
4012	021676	051524	030440	026465	
4013	021704	021467			
4014	021706	044445	042524	035122	MSG16: .ASCII /%ITER: #/
4015	021714	021440			
4016	021716	052045	020103	044502	MSG18: .ASCII /%TC BITS 12-0 #/
4017	021724	051524	030440	026462	
4018	021732	020060	043		
4019	021735	045	041506	041040	MSG19: .ASCII /%FC BITS 15-0 #/
4020	021742	052111	020123	032461	
4021	021750	030055	021440		
4022	021754	043045	047125	041440	MSG20: .ASCII /%FUN CODE BITS 5-1 OF C1 #/
4023	021762	042117	020105	044502	
4024	021770	051524	032440	030455	
4025	021776	047440	020106	030503	
4026	022004	021440			
4027	022006	043445	020117	044502	MSG21: .ASCII /%GO BIT NOT CORRECT AT START #/
4028	022014	020124	047516	020124	
4029	022022	047503	051122	041505	
4030	022030	020124	052101	051440	
4031	022036	040524	052122	021440	
4032	022044	043445	020117	044502	MSG22: .ASCII /%GO BIT NOT SET #/
4033	022052	020124	047516	020124	
4034	022060	042523	020124	043	
4035	022065	045	047507	041040	MSG23: .ASCII /%GO BIT NOT RESET BY INIT #/

4036	022072	052111	047040	052117	
4037	022100	051040	051505	052105	
4038	022106	041040	020131	047111	
4039	022114	052111	021440		
4040	022120	042045	054522	047040	MSG24: .ASCII /%DRY NOT SET BY INIT #/
4041	022126	052117	051440	052105	
4042	022134	041040	020131	047111	
4043	022142	052111	021440		
4044	022146	042045	054522	047040	MSG25: .ASCII /%DRY NOT RESET BY GO=1#/
4045	022154	052117	051040	051505	
4046	022162	052105	041040	020131	
4047	022170	047507	030475	043	
4048	022175	045	051104	020131	MSG25A: .ASCII /%DRY NOT SET BY GO=0#/
4049	022202	047516	020124	042523	
4050	022210	020124	054502	043440	
4051	022216	036517	021460		
4052	022222	047045	020117	047111	MSG26: .ASCII /%NO INTERRUPT RETURNED#/
4053	022230	042524	051122	050125	
4054	022236	020124	042522	052524	
4055	022244	047122	042105	043	
4056	022251	045	040502	020104	MSG27: .ASCII /%BAD STATUS#/
4057	022256	052123	052101	051525	
4058	022264	043			
4059	022265	040	047123	020072	MSG30: .ASCII / SN: #/
4060	022272	043			
4061	022273	045	051105	020122	MSG31: .ASCII /%ERR NOT SET #/
4062	022300	047516	020124	042523	
4063	022306	020124	043		
4064	022311	045	052101	020101	MSG32: .ASCII /%ATA NOT SET #/
4065	022316	047516	020124	042523	
4066	022324	020124	043		
4067	022327	045	051501	041040	MSG33: .ASCII /%AS BIT NOT SET #/
4068	022334	052111	047040	052117	
4069	022342	051440	052105	021440	
4070	022350	051445	020103	047516	MSG34: .ASCII /%SC NOT SET #/
4071	022356	020124	042523	020124	
4072	022364	043			
4073	022365	045	051124	020105	MSG35: .ASCII /%TRE NOT SET #/
4074	022372	047516	020124	042523	
4075	022400	020124	043		
4076	022403	045	046123	020101	MSG36: .ASCII /%SLA NOT SET #/
4077	022410	047516	020124	042523	
4078	022416	020124	043		
4079	022421	045	051523	020103	MSG37: .ASCII /%SSC NOT SET #/
4080	022426	047516	020124	042523	
4081	022434	020124	043		
4082	022437	040	020077	043	MSG40: .ASCII / ? #/
4083	022443	045	042445	042116	MSG41: .ASCII /%END OF PASS #/
4084	022450	047440	020106	040520	
4085	022456	051523	021440		
4086	022462	042045	040505	020104	MSG42: .ASCII /%DEAD TRACK #/
4087	022470	051124	041501	020113	
4088	022476	043			
4089	022477	045	046445	047101	MSG43: .ASCII /%MANUAL TESTS (14-17) INHIBITED: HALT#/
4090	022504	040525	020114	042524	
4091	022512	052123	020123	030450	

4092	022520	026464	033461	020051	
4093	022526	047111	044510	044502	
4094	022534	042524	035104	044040	
4095	022542	046101	022524		
4096	022546	042522	042523	042514	.ASCII /RESELECT AND PRESS CONTINUE%/
4097	022554	052103	040440	042116	
4098	022562	050040	042522	051523	
4099	022570	041440	047117	044524	
4100	022576	052516	022505	043	
4101	022603	045	042522	044507	MSG44: .ASCII /%REGISTER START: #/
4102	022610	052123	051105	051440	
4103	022616	040524	052122	020072	
4104	022624	043			
4105	022625	045	042526	052103	MSG45: .ASCII /%VECTOR ADDRESS: #/
4106	022632	051117	040440	042104	
4107	022640	042522	051523	020072	
4108	022646	043			
4109	022647	045	051503	020061	MSG46: .ASCII /%CS1 WC BA FC CS2 DS ER AS/
4110	022654	020040	053440	020103	
4111	022662	020040	020040	040502	
4112	022670	020040	020040	043040	
4113	022676	020103	020040	020040	
4114	022704	051503	020062	020040	
4115	022712	042040	020123	020040	
4116	022720	020040	051105	020040	
4117	022726	020040	040440	123	
4118	022733	040	020040	020040	.ASCII / MR TC%/
4119	022740	051115	020040	020040	
4120	022746	052040	022503	043	
4121	022753	045	047516	020124	MSG47: .ASCII /%NOT RESET BY DRIVE CLEAR%/
4122	022760	042522	042523	020124	
4123	022766	054502	042040	044522	
4124	022774	042526	041440	042514	
4125	023002	051101	043		
4126	023005	045	046101	044120	MSG50: .ASCII /%ALPHA NOT SET%/
4127	023012	020101	047516	020124	
4128	023020	042523	021524		
4129	023024	052445	042516	050130	MSG51: .ASCII /%UNEXPECTED ERROR BITS%/
4130	023032	041505	042524	020104	
4131	023040	051105	047522	020122	
4132	023046	044502	051524	043	
4133	023053	045	040502	020104	MSG53: .ASCII /%BAD LRC #/
4134	023060	051114	020103	043	
4135	023065	045	040502	020104	MSG54: .ASCII /%BAD CK #/
4136	023072	045503	021440		
4137	023076	051445	052105	050125	MSG55: .ASCII /%SETUP ERROR: CHECK WRAP 0 WITH CONTROL LOGIC TEST II TEST 7%/
4138	023104	042440	051122	051117	
4139	023112	020072	044103	041505	
4140	023120	020113	051127	050101	
4141	023126	030040	053440	052111	
4142	023134	020110	047503	052116	
4143	023142	047522	020114	047514	
4144	023150	044507	020103	042524	
4145	023156	052123	044440	020111	
4146	023164	042524	052123	033440	
4147	023172	043			

4148	023173	045	052123	052101	MSG56: .ASCII /%STATIC TESTS ONLY: #/
4149	023200	041511	052040	051505	
4150	023206	051524	047440	046116	
4151	023214	035131	021440		
4152	023220	052045	047515	020063	MSG57: .ASCII /%TM03 DRIVE: #/
4153	023226	051104	053111	035105	
4154	023234	021440			
4155	023236	044445	020123	047503	MS57A: .ASCII /%IS CONTROLLER JUMPERED IN NON-STANDARD MODE, /<15><12>
4156	023244	052116	047522	046114	
4157	023252	051105	045040	046525	
4158	023260	042520	042522	020104	
4159	023266	047111	047040	047117	
4160	023274	051455	040524	042116	
4161	023302	051101	020104	047515	
4162	023310	042504	006454	012	
4163	023315	124	050131	020105	.ASCII /TYPE 2 FOR NON=STANDARD OR CR FOR STANDARD ? #/
4164	023322	020062	047506	020122	
4165	023330	047516	036516	052123	
4166	023336	047101	040504	042122	
4167	023344	047440	020122	051103	
4168	023352	043040	051117	051440	
4169	023360	040524	042116	051101	
4170	023366	020104	020077	020040	
4171	023374	020040	043		
4172	023377	045	042524	033061	MSG58: .ASCII '%TE16/TU77 SLAVE: #'; ++B
4173	023404	052057	033525	020067	
4174	023412	046123	053101	035105	
4175	023420	021440			
4176	023422	021445			MSG59: .ASCII /%#/
4177	023424	021534			MSG60: .ASCII /\#/
4178	023426	051045	046505	053117	MSG62: .ASCII /%REMOVE TMDP FROM SLAVE TO BE TESTED%/
4179	023434	020105	046524	050104	
4180	023442	043040	047522	020115	
4181	023450	046123	053101	020105	
4182	023456	047524	041040	020105	
4183	023464	042524	052123	042105	
4184	023472	021445			
4185	023474	044045	051101	053504	MSG63: .ASCII /%HARDWARE SWR IN USE%/
4186	023502	051101	020105	053523	
4187	023510	020122	047111	052440	
4188	023516	042523	021445		
4189	023522	051445	040514	042526	MSG64: .ASCII /%SLAVE TYPE: #/ ; ++B
4190	023530	052040	050131	035105	
4191	023536	021440			
4192	023540	052524	033467	043	MSG65: .ASCII /TU77#/ ; ++B
4193	023545	124	030505	021466	MSG66: .ASCII /TE16#/ ; ++B
4194	023552	051445	040514	042526	MSG67: .ASCII /%SLAVE TYPE (0=TE16,1=TU77): #/ ; ++B
4195	023560	052040	050131	020105	
4196	023566	030050	052075	030505	
4197	023574	026066	036461	052524	
4198	023602	033467	035051	021440	
4199	023610	022445	047111	047503	MSG68: .ASCII /%%INCORRECT SLAVE TYPE!!! PROGRAM ABORTED#/ ; ++B
4200	023616	051122	041505	020124	
4201	023624	046123	053101	020105	
4202	023632	054524	042520	020441	
4203	023640	020041	051120	043517	

4204	023646	040522	020115	041101	
4205	023654	051117	042524	021504	
4206	023662	046111	042514	040507	MSG69: .ASCII /ILLEGAL#/ ;++B
4207	023670	021514			
4208					:TEST HEADER*****
4209					
4210	023672	022445	047514	044507	MSLT1: .ASCII /%%LOGIC TEST 1: DRIVE ADDRESSING (M8909 RH)#/
4211	023700	020103	042524	052123	
4212	023706	030440	020072	051104	
4213	023714	053111	020105	042101	
4214	023722	051104	051505	044523	
4215	023730	043516	024040	034115	
4216	023736	030071	020071	044122	
4217	023744	021451			
4218	023746	022445	047514	044507	MSLT2: .ASCII /%%LOGIC TEST 2: REGISTER ADDRESSING (M8909 RH)#/
4219	023754	020103	042524	052123	
4220	023762	031040	020072	042522	
4221	023770	044507	052123	051105	
4222	023776	040440	042104	042522	
4223	024004	051523	047111	020107	
4224	024012	046450	034470	034460	
4225	024020	051040	024510	043	
4226	024025	045	046045	043517	MSLT3: .ASCII /%%LOGIC TEST 3: CONTROL BUS TEST (RH M8905-YB M8909)#/
4227	024032	041511	052040	051505	
4228	024040	020124	035063	041440	
4229	024046	047117	051124	046117	
4230	024054	041040	051525	052040	
4231	024062	051505	020124	051050	
4232	024070	020110	034115	030071	
4233	024076	026465	041131	046440	
4234	024104	034470	034460	021451	
4235	024112	022445	047514	044507	MSLT4: .ASCII /%%LOGIC TEST 4: SLAVE ADDRESSING (M8905-YB M8933)#/
4236	024120	020103	042524	052123	
4237	024126	032040	020072	046123	
4238	024134	053101	020105	042101	
4239	024142	051104	051505	044523	
4240	024150	043516	024040	034115	
4241	024156	030071	026465	041131	
4242	024164	046440	034470	031463	
4243	024172	021451			
4244	024174	022445	047514	044507	MSLT5: .ASCII /%%LOGIC TEST 5: MR BIT TEST (M8905-YB)#/
4245	024202	020103	042524	052123	
4246	024210	032440	020072	051115	
4247	024216	041040	052111	052040	
4248	024224	051505	020124	046450	
4249	024232	034470	032460	054455	
4250	024240	024502	043		
4251	024243	045	046045	043517	MSLT6: .ASCII /%%LOGIC TEST 6: TC BIT TEST (M8905-YB)#/
4252	024250	041511	052040	051505	
4253	024256	020124	035066	052040	
4254	024264	020103	044502	020124	
4255	024272	042524	052123	024040	
4256	024300	034115	030071	026465	
4257	024306	041131	021451		
4258	024312	022445	047514	044507	MSLT7: .ASCII /%%LOGIC TEST 7: FC BIT TEST (M8905-YB)#/
4259	024320	020103	042524	052123	

4260	024326	033440	020072	041506
4261	024334	041040	052111	052040
4262	024342	051505	020124	046450
4263	024350	034470	032460	054455
4264	024356	024502	043	
4265	024361	045	046045	043517
4266	024366	041511	052040	051505
4267	024374	020124	030061	020072
4268	024402	052506	041516	044524
4269	024410	047117	041040	052111
4270	024416	052040	051505	020124
4271	024424	046450	034470	032460
4272	024432	054455	024502	043
4273	024437	045	046045	043517
4274	024444	041511	052040	051505
4275	024452	020124	030461	020072
4276	024460	047507	041040	052111
4277	024466	052040	051505	020124
4278	024474	046450	034470	034460
4279	024502	021451		
4280	024504	022445	047514	044507
4281	024512	020103	042524	052123
4282	024520	030440	035062	042040
4283	024526	044522	042526	051040
4284	024534	040505	054504	041040
4285	024542	052111	024040	034115
4286	024550	030071	024471	043
4287	024555	045	046045	043517
4288	024562	041511	052040	051505
4289	024570	020124	031461	020072
4290	024576	047111	042524	051122
4291	024604	050125	020124	042524
4292	024612	052123	024040	044122
4293	024620	021451		
4294	024622	022445	047514	044507
4295	024630	020103	042524	052123
4296	024636	030440	035064	051440
4297	024644	040524	052524	020123
4298	024652	052101	041040	052117
4299	024660	047454	026516	044514
4300	024666	042516	053454	044522
4301	024674	042524	050040	047522
4302	024702	042524	052103	042105
4303	024710	024040	047516	053440
4304	024716	044522	042524	051040
4305	024724	047111	024507	043
4306	024731	045	046045	043517
4307	024736	041511	052040	051505
4308	024744	020124	032461	020072
4309	024752	052123	052101	051525
4310	024760	040440	020124	047502
4311	024766	026124	043117	026506
4312	024774	044514	042516	053454
4313	025002	044522	042524	050040
4314	025010	047522	042524	052103
4315	025016	042105	043	

MSLT10: .ASCII /%%LOGIC TEST 10: FUNCTION BIT TEST (M8905-YB)#/

MSLT11: .ASCII /%%LOGIC TEST 11: GO BIT TEST (M8909)#/

MSLT12: .ASCII /%%LOGIC TEST 12: DRIVE READY BIT (M8909)#/

MSLT13: .ASCII /%%LOGIC TEST 13: INTERRUPT TEST (RH)#/

MSLT14: .ASCII /%%LOGIC TEST 14: STATUS AT BOT,ON-LINE,WRITE PROTECTED (NO WRITE RING)#/

MSLT15: .ASCII /%%LOGIC TEST 15: STATUS AT BOT,OFF-LINE,WRITE PROTECTED#/

4316	025021	045	046045	043517	MSLT16: .ASCII /%%LOGIC TEST 16: STATUS AT EOT,ON-LINE,WRITE PROTECTED#/
4317	025026	041511	052040	051505	
4318	025034	020124	033061	020072	
4319	025042	052123	052101	051525	
4320	025050	040440	020124	047505	
4321	025056	026124	047117	046055	
4322	025064	047111	026105	051127	
4323	025072	052111	020105	051120	
4324	025100	052117	041505	042524	
4325	025106	021504			
4326	025110	022445	047514	044507	MSLT17: .ASCII /%%LOGIC TEST 17: STATUS AT ON-LINE,WRITE ENABLED#/
4327	025116	020103	042524	052123	
4328	025124	030440	035067	051440	
4329	025132	040524	052524	020123	
4330	025140	052101	047440	026516	
4331	025146	044514	042516	053454	
4332	025154	044522	042524	042440	
4333	025162	040516	046102	042105	
4334	025170	043			
4335	025171	045	046045	043517	MSLT20: .ASCII /%%LOGIC TEST 20: ILLEGAL FUNCTION TEST (M8909)#/
4336	025176	041511	052040	051505	
4337	025204	020124	030062	020072	
4338	025212	046111	042514	040507	
4339	025220	020114	052506	041516	
4340	025226	044524	047117	052040	
4341	025234	051505	020124	046450	
4342	025242	034470	034460	021451	
4343	025250	022445	047514	044507	MSLT21: .ASCII /%%LOGIC TEST 21: RMR(M8909)#/
4344	025256	020103	042524	052123	
4345	025264	031040	035061	051040	
4346	025272	051115	046450	034470	
4347	025300	034460	021451		
4348	025304	022445	047514	044507	MSLT22: .ASCII /%%LOGIC TEST 22: CPAR(M8909)#/
4349	025312	020103	042524	052123	
4350	025320	031040	035062	041440	
4351	025326	040520	024122	034115	
4352	025334	030071	024471	043	
4353	025341	045	046045	043517	MSLT23: .ASCII /%%LOGIC TEST 23: FMT(M8905-YB M8906)#/
4354	025346	041511	052040	051505	
4355	025354	020124	031462	020072	
4356	025362	046506	024124	034115	
4357	025370	030071	026465	041131	
4358	025376	046440	034470	033060	
4359	025404	021451			
4360	025406	022445	047514	044507	MSLT24: .ASCII /%%LOGIC TEST 24: DPAR(M8906 RH)#/
4361	025414	020103	042524	052123	
4362	025422	031040	035064	042040	
4363	025430	040520	024122	034115	
4364	025436	030071	020066	044122	
4365	025444	021451			
4366	025446	022445	047514	044507	MSLT25: .ASCII /%%LOGIC TEST 25: NEF(M8909)#/
4367	025454	020103	042524	052123	
4368	025462	031040	035065	047040	
4369	025470	043105	046450	034470	
4370	025476	034460	021451		
4371	025502	022445	047514	044507	MSLT26: .ASCII /%%LOGIC TEST 26: FCE(M8909)#/

4372	025510	020103	042524	052123	
4373	025516	031040	035066	043040	
4374	025524	042503	046450	034470	
4375	025532	034460	021451		
4376	025536	022445	047514	044507	MSLT27: .ASCII /%%LOGIC TEST 27: ILR(M8909)##/
4377	025544	020103	042524	052123	
4378	025552	031040	035067	044440	
4379	025560	051114	046450	034470	
4380	025566	034460	021451		
4381	025572	022445	047514	044507	MSLT30: .ASCII /%%LOGIC TEST 30:DTE(M8906 RH)##/
4382	025600	020103	042524	052123	
4383	025606	031440	035060	052104	
4384	025614	024105	034115	030071	
4385	025622	020066	044122	021451	
4386	025630	022445	047514	044507	MSLT31: .ASCII /%%LOGIC TEST 31: OPI(M8933)##/
4387	025636	020103	042524	052123	
4388	025644	031440	035061	047440	
4389	025652	044520	046450	034470	
4390	025660	031463	021451		
4391	025664	022445	047514	044507	MSLT32: .ASCII /%%LOGIC TEST 32: UNS(M8909)##/
4392	025672	020103	042524	052123	
4393	025700	031440	035062	052440	
4394	025706	051516	046450	034470	
4395	025714	034460	021451		
4396	025720	022445	047514	044507	MSLT33: .ASCII /%%LOGIC TEST 33: PIP(M8909)##/
4397	025726	020103	042524	052123	
4398	025734	031440	035063	050040	
4399	025742	050111	046450	034470	
4400	025750	034460	021451		
4401	025754	022445	047514	044507	MSLT34: .ASCII /%%LOGIC TEST 34: PES(M8931)##/
4402	025762	020103	042524	052123	
4403	025770	031440	035064	050040	
4404	025776	051505	046450	034470	
4405	026004	030463	021451		
4406	026010	022445	047514	044507	MSLT35: .ASCII /%%LOGIC TEST 35: SAC(M8933 M8905-YB)##/
4407	026016	020103	042524	052123	
4408	026024	031440	035065	051440	
4409	026032	041501	046450	034470	
4410	026040	031463	046440	034470	
4411	026046	032460	054455	024502	
4412	026054	043			
4413	026055	045	046045	043517	MSLT36: .ASCII /%%LOGIC TEST 36: FCS(M8933 M8905-YB)##/
4414	026062	041511	052040	051505	
4415	026070	020124	033063	020072	
4416	026076	041506	024123	034115	
4417	026104	031471	020063	034115	
4418	026112	030071	026465	041131	
4419	026120	021451			
4420	026122	022445	047514	044507	MSLT37: .ASCII /%%LOGIC TEST 37: ACCL(M8933 M8905-YB)##/
4421	026130	020103	042524	052123	
4422	026136	031440	035067	040440	
4423	026144	041503	024114	034115	
4424	026152	031471	020063	034115	
4425	026160	030071	026465	041131	
4426	026166	021451			
4427	026170	022445	047514	044507	MSLT40: .ASCII /%%LOGIC TEST 40: PE TAPE MARK(M8932)##/

4428	026176	020103	042524	052123	
4429	026204	032040	035060	050040	
4430	026212	020105	040524	042520	
4431	026220	046440	051101	024113	
4432	026226	034115	031471	024462	
4433	026234	043			
4434	026235	045	046045	043517	MSLT41: .ASCII /%%LOGIC TEST 41: NRZ TAPE MARK (M8934)##/
4435	026242	041511	052040	051505	
4436	026250	020124	030464	020072	
4437	026256	051116	020132	040524	
4438	026264	042520	046440	051101	
4439	026272	020113	046450	034470	
4440	026300	032063	021451		
4441	026304	022445	047514	044507	MSLT42: .ASCII /%%LOGIC TEST 42: CRC(M8934)##/
4442	026312	020103	042524	052123	
4443	026320	032040	035062	041440	
4444	026326	041522	046450	034470	
4445	026334	032063	021451		
4446	026340	022445	047514	044507	MSLT43: .ASCII /%%LOGIC TEST 43: LRC(M8934)##/
4447	026346	020103	042524	052123	
4448	026354	032040	035063	046040	
4449	026362	041522	046450	034470	
4450	026370	032063	021451		
4451	026374	022445	047514	044507	MSLT44: .ASCII /%%LOGIC TEST 44: CORRECTABLE DATA (M8932 M8901)##/
4452	026402	020103	042524	052123	
4453	026410	032040	035064	041440	
4454	026416	051117	042522	052103	
4455	026424	041101	042514	042040	
4456	026432	052101	020101	046450	
4457	026440	034470	031063	046440	
4458	026446	034470	030460	021451	
4459	026454	022445	047514	044507	MSLT45: .ASCII /%%LOGIC TEST 45: INCORRECTABLE DATA (M8932 M8934)##/
4460	026462	020103	042524	052123	
4461	026470	032040	035065	044440	
4462	026476	041516	051117	042522	
4463	026504	052103	041101	042514	
4464	026512	042040	052101	020101	
4465	026520	046450	034470	031063	
4466	026526	046440	034470	032063	
4467	026534	021451			
4468	026536	022445	047514	044507	MSLT46: .ASCII /%%LOGIC TEST 46: PEF(M8932)##/
4469	026544	020103	042524	052123	
4470	026552	032040	035066	050040	
4471	026560	043105	046450	034470	
4472	026566	031063	021451		
4473	026572	022445	047514	044507	MSLT47: .ASCII /%%LOGIC TEST 47: FC OVERFLOW (M8905-YB)##/
4474	026600	020103	042524	052123	
4475	026606	032040	035067	043040	
4476	026614	020103	053117	051105	
4477	026622	046106	053517	024040	
4478	026630	034115	030071	026465	
4479	026636	041131	021451		
4480	026642	022445	047514	044507	MSLT50: .ASCII /%%LOGIC TEST 50: NEF WHEN WRITE PE ON NRZ SLAVE##/
4481	026650	020103	042524	052123	
4482	026656	032440	035060	047040	
4483	026664	043105	053440	042510	

4484	026672	020116	051127	052111
4485	026700	020105	042520	047440
4486	026706	020116	051116	020132
4487	026714	046123	053101	021505
4488	026722	022445	047514	044507
4489	026730	020103	042524	052123
4490	026736	032440	035061	047040
4491	026744	043105	053440	042510
4492	026752	020116	051127	052111
4493	026760	020105	051116	020132
4494	026766	047117	050040	020105
4495	026774	046123	053101	021505

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

```
4496
4497
4498
4499 027002 052045 050131 020105 MMSG0: .ASCII /%TYPE CR WHEN READY;#/
4500 027010 051103 053440 042510
4501 027016 020116 042522 042101
4502 027024 035531 043
4503 027027 045 046445 052517 MMSG1: .ASCII /%%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE:#/
4504 027034 052116 052040 050101
4505 027042 020105 044527 044124
4506 027050 047040 020117 051127
4507 027056 052111 020105 044522
4508 027064 043516 020054 047514
4509 027072 042101 052040 020117
4510 027100 047502 026124 051440
4511 027106 052105 052040 020117
4512 027114 047117 046040 047111
4513 027122 035105 043
4514 027125 045 042523 020124 MMSG2: .ASCII /%SET TO OFFLINE:#/
4515 027132 047524 047440 043106
4516 027140 044514 042516 021472
4517 027146 046445 053117 020105 MMSG3: .ASCII /%MOVE FORWARD TO EOT, ONLINE:#/
4518 027154 047506 053522 051101
4519 027162 020104 047524 042440
4520 027170 052117 020054 047117
4521 027176 044514 042516 021472
4522 027204 047445 043106 046040 MMSG4: .ASCII /%OFF LINE REVERSE PAST EOT, INSERT WRITE RING, ON LINE#/
4523 027212 047111 020105 042522
4524 027220 042526 051522 020105
4525 027226 040520 052123 042440
4526 027234 052117 020054 047111
4527 027242 042523 052122 053440
4528 027250 044522 042524 051040
4529 027256 047111 026107 047440
4530 027264 020116 044514 042516
4531 027272 043
4532 027273 045 046445 053117 MMSG5: .ASCII /%%MOVE TAPE TO BOT; ON LINE#/
4533 027300 020105 040524 042520
4534 027306 052040 020117 047502
4535 027314 035524 047440 020116
4536 027322 044514 042516 043
```

```
4537
4538 ;TAG MESSAGE
4539
4540 027327 045 053523 020122 $MSWR: .ASCII /%SWR = #/
4541 027334 020075 043
4542 027337 040 042516 020127 $MNEW: .ASCII / NEW = #/
4543 027344 020075 043
4544 027347 045 046123 020101 TMS1: .ASCII /%SLA #/
4545 027354 043
4546 027355 045 047502 020124 TMS2: .ASCII /%BOT #/
4547 027362 043
4548 027363 045 046524 021440 TMS3: .ASCII /%TM #/
4549 027370 044445 041104 021440 TMS4: .ASCII /%IDB #/
4550 027376 051445 053504 020116 TMS5: .ASCII /%SDWN #/
4551 027404 043
4552 027405 045 042520 020123 TMS6: .ASCII /%PES #/
4553 027412 043
4554 027413 045 051523 020103 TMS7: .ASCII /%SSC #/
4555 027420 043
4556 027421 045 051104 020131 TMS8: .ASCII /%DRY #/
4557 027426 043
4558 027427 045 050104 020122 TMS9: .ASCII /%DPR #/
4559 027434 043
4560 027435 045 052116 020114 TMS10: .ASCII /%NTL #/
4561 027442 043
4562 027443 045 047505 020124 TMS11: .ASCII /%EOT #/
4563 027450 043
4564 027451 045 051127 020114 TMS12: .ASCII /%WRL #/
4565 027456 043
4566 027457 045 047515 020114 TMS13: .ASCII /%MOL #/
4567 027464 043
4568 027465 045 044520 020120 TMS14: .ASCII /%PIP #/
4569 027472 043
4570 027473 045 051105 020122 TMS15: .ASCII /%ERR #/
4571 027500 043
4572 027501 045 052101 020101 TMS16: .ASCII /%ATA #/
4573 027506 043
4574 027507 045 046111 020106 TMS17: .ASCII /%ILF #/
4575 027514 043
4576 027515 045 046111 020122 TMS18: .ASCII /%ILR #/
4577 027522 043
4578 027523 045 046522 020122 TMS19: .ASCII /%RMR #/
4579 027530 043
4580 027531 045 050103 051101 TMS20: .ASCII /%CPAR #/
4581 027536 021440
4582 027540 043045 052115 021440 TMS21: .ASCII /%FMT #/
4583 027546 042045 040520 020122 TMS22: .ASCII /%DPAR #/
4584 027554 043
4585 027555 045 047111 020103 TMS23: .ASCII /%INC #/
4586 027562 043
4587 027563 045 050126 020105 TMS24: .ASCII /%VPE #/
4588 027570 043
4589 027571 045 042520 020106 TMS25: .ASCII /%PEF #/
4590 027576 043
4591 027577 045 051114 020103 TMS26: .ASCII /%LRC #/
4592 027604 043
```


4649	030100	177777	-1
4650	030102	177777	-1
4651	030104	177777	-1
4652	030106	177777	-1
4653	030110	177777	-1
4654	030112	177777	-1
4655	030114	177777	-1
4656	030116	177777	-1
4657	030120	177777	-1
4658	030122	177777	-1
4659	030124	177777	-1
4660	030126	177777	-1
4661	030130	177777	-1
4662	030132	177777	-1
4663	030134	177777	-1
4664	030136	177777	-1
4665	030140	177777	-1
4666	030142	177777	-1
4667	030144	177777	-1
4668	030146	177777	-1
4669	030150	177777	-1
4670	030152	177777	-1
4671	030154	177777	-1
4672	030156	177777	-1
4673	030160	177777	-1
4674	030162	177777	-1
4675	030164	177777	-1
4676	030166	177777	-1
4677	030170	177777	-1
4678	030172	177777	-1
4679	030174	177777	-1
4680	030176	177777	-1
4681	030200	177777	-1
4682	030202	177777	-1
4683	030204	177777	-1
4684	030206	177777	-1
4685	030210	177777	-1
4686	030212	177777	-1
4687	030214	177777	-1
4688	030216	177777	-1
4689	030220	177777	-1
4690	030222	177777	-1
4691	030224	177777	-1
4692	030226	177777	-1
4693	030230	177777	-1
4694	030232	177777	-1
4695	030234	177777	-1
4696	030236	177777	-1
4697			
4698			
4699			;READ BUFFER
4700			
4701	030240	000100	
4702	030240	000000	RDATA: 0
4703	030242	000000	0
4704	030244	000000	0

4705	030246	000000	0
4706	030250	000000	0
4707	030252	000000	0
4708	030254	000000	0
4709	030256	000000	0
4710	030260	000000	0
4711	030262	000000	0
4712	030264	000000	0
4713	030266	000000	0
4714	030270	000000	0
4715	030272	000000	0
4716	030274	000000	0
4717	030276	000000	0
4718	030300	000000	0
4719	030302	000000	0
4720	030304	000000	0
4721	030306	000000	0
4722	030310	000000	0
4723	030312	000000	0
4724	030314	000000	0
4725	030316	000000	0
4726	030320	000000	0
4727	030322	000000	0
4728	030324	000000	0
4729	030326	000000	0
4730	030330	000000	0
4731	030332	000000	0
4732	030334	000000	0
4733	030336	000000	0
4734	030340	000000	0
4735	030342	000000	0
4736	030344	000000	0
4737	030346	000000	0
4738	030350	000000	0
4739	030352	000000	0
4740	030354	000000	0
4741	030356	000000	0
4742	030360	000000	0
4743	030362	000000	0
4744	030364	000000	0
4745	030366	000000	0
4746	030370	000000	0
4747	030372	000000	0
4748	030374	000000	0
4749	030376	000000	0
4750	030400	000000	0
4751	030402	000000	0
4752	030404	000000	0
4753	030406	000000	0
4754	030410	000000	0
4755	030412	000000	0
4756	030414	000000	0
4757	030416	000000	0
4758	030420	000000	0
4759	030422	000000	0
4760	030424	000000	0

4761 030426 000000 0
4762 030430 000000 0
4763 030432 000000 0
4764 030434 000000 0
4765 030436 000000 0

;WRAP AROUND MESSAGES*****

4769 030440 042045 052101 020101 WMSG27: .ASCII /%DATA PAT:#/

4770	030446	040520	035124	043	
4771	030453	045	047505	020122	WMSG31: .ASCII /%EOR CLEAR DID NOT CLEAR GO%#/ 4772 030460 046103 040505 020122
4773	030466	044504	020104	047516	
4774	030474	020124	046103	040505	
4775	030502	020122	047507	021445	

4776
4777
4778 030510 000000 PRE: .EVEN 0
4779 030512 000000 0
4780 030514 000000 0
4781 030516 000000 0
4782 030520 000000 0
4783 030522 000000 0
4784 030524 000000 0
4785 030526 000000 0
4786 030530 000000 0
4787 030532 000000 0
4788 030534 000000 0
4789 030536 000000 0
4790 030540 000000 0
4791 030542 000000 0
4792 030544 000000 0
4793 030546 000000 0
4794 030550 000000 0
4795 030552 000000 0
4796 030554 000000 0
4797 030556 000000 0
4798 030560 000000 0
4799 030562 000000 0
4800 030564 000000 0
4801 030566 000000 0
4802 030570 000000 0
4803 030572 000000 0
4804 030574 000000 0
4805 030576 000000 0
4806 030600 000000 0
4807 030602 000000 0
4808 030604 000000 0
4809 030606 000000 0
4810 030610 000000 0
4811 030612 000000 0
4812 030614 000000 0
4813 030616 000000 0
4814 030620 000000 0
4815 030622 000000 0
4816 030624 000000 0

4817	030626	000000		0
4818	030630	000000		0
4819	030632	000000	POST:	0
4820	030634	000000		0
4821	030636	000000		0
4822	030640	000000		0
4823	030642	000000		0
4824	030644	000000		0
4825	030646	000000		0
4826	030650	000000		0
4827	030652	000000		0
4828	030654	000000		0
4829	030656	000000		0
4830	030660	000000		0
4831	030662	000000		0
4832	030664	000000		0
4833	030666	000000		0
4834	030670	000000		0
4835	030672	000000		0
4836	030674	000000		0
4837	030676	000000		0
4838	030700	000000		0
4839	030702	000000		0
4840	030704	000000		0
4841	030706	000000		0
4842	030710	000000		0
4843	030712	000000		0
4844	030714	000000		0
4845	030716	000000		0
4846	030720	000000		0
4847	030722	000000		0
4848	030724	000000		0
4849	030726	000000		0
4850	030730	000000		0
4851	030732	000000		0
4852	030734	000000		0
4853	030736	000000		0
4854	030740	000000		0
4855	030742	000000		0
4856	030744	000000		0
4857	030746	000000		0
4858	030750	000000		0
4859	030752	000000		0
4860	030754	000000	WBUFF:	0
4861		031366		.=.+410
4862	031366	000000	RBUFF:	0
4863				
4864		000001		.END

LT15A	005620	2172	2176#	
LT15IT	005636	1473	2179#	2184
LT15X	005676	2183	2187#	
LT15XX	005702	2174	2188#	
LT16	005706	1474	2192#	
LT16A	005730	2193	2197#	
LT16IT	005746	1475	2200#	2205
LT16X	006006	2204	2208#	
LT16XX	006012	2195	2209#	
LT17	006020	1476	2217#	
LT17A	006042	2218	2222#	
LT17IT	006060	1477	2225#	2230
LT17X	006120	2229	2233#	
LT17XX	006124	2220	2234#	
LT2	003210	1450	1451	1799#
LT2A	003250	1806#	1816	1831
LT2B	003270	1809	1811#	
LT2C	003304	1812	1814#	
LT2ERG	003364	1821	1824	1827#
LT2ER1	003314	1810	1819#	
LT2ER2	003332	1813	1822#	
LT2ER3	003350	1825#		
LT2IT	003212	1800#		
LT2LP	003400	1827	1830#	
LT2X	003404	1817	1832#	
LT20	006132	1478	2246#	
LT20A	006160	2247	2250#	2267
LT20B	006236	2256	2261#	
LT20C	006246	2260	2262	2264#
LT20IT	006146	1479	2248#	
LT20X	006260	2265	2268#	
LT21	006274	1480	2274#	
LT21A	006400	2284	2289#	
LT21B	006410	2288	2290	2293#
LT21IT	006310	1481	2275	2276#
LT21XA	006414	2294#	2295	
LT22	006440	1482	2303#	
LT22A	006540	2312	2317#	
LT22IT	006454	1483	2304	2305#
LT22X	006550	2316	2318	2320#
LT23	006564	1484	2326#	
LT23A	006664	2336	2341#	
LT23IT	006600	1485	2327	2328#
LT23X	006674	2340	2342	2344#
LT24	006714	1486	2350#	
LT24B	007042	2369#	2375	
LT24B0	007064	2372	2374#	
LT24C	007076	2378#	2379	
LT24D	007160	2381	2392#	
LT24IT	006730	1487	2351	2352#
LT24X	007204	2391	2396	2398#
LT25	007240	1488	2407#	
LT25A	007346	2416	2422#	
LT25IT	007246	1489	2408#	2417
LT25X	007356	2421	2423	2425#
LT26	007372	1490	2431#	

LT261T	007400	1491	2432#	2456		
LT26X	007572	2460	2463	2465#		
LT27	007606	1492	2471#			
LT27A	007642	2473	2477#	2487		
LT27B	007700	2480	2484#			
LT271T	007632	1493	2475#			
LT27X	007710	2485	2488#			
LT27XX	007720	2472	2490#			
LT3	003414	1452	1836#			
LT3A	003434	1840#	1850	1852	1864	
LT3B	003454	1844#	1855	1857		
LT3C	003470	1847#	1859			
LT3ER1	003500	1843	1851#			
LT3ER2	003526	1846	1856#			
LT31T	003416	1453	1837#			
LT3X	003550	1848	1860#			
LT3XX	003566	1861	1865#			
LT30	007724	1494	2494#			
LT30A	010026	2504	2507#			
LT30B	010070	2514#	2517			
LT30C	010130	2515	2524#			
LT30D	010146	2527#	2528			
LT30E	010176	2530	2534#			
LT301T	007746	1495	2496	2497#		
LT30X	010222	2523	2533	2538	2540#	
LT31	010242	1496	2548#			
LT31A	010420	2574	2576#			
LT311T	010250	1497	2549#			
LT31X	010524	2570	2572	2593	2595#	2643
LT32	011072	1498	2695#			
LT321T	011106	1499	2696	2697#		
LT32X	011234	2716	2718	2720#		
LT32XX	011244	2708	2722#			
LT33	011250	1500	2731#			
LT331T	011264	1501	2732	2733#		
LT33X	011342	2738	2742#			
LT34	011352	1502	2747#			
LT34A	011412	2753#				
LT34A1	011372	2750#	2756			
LT34B	011442	2754	2758#			
LT34C	011446	2759#	2761			
LT341T	011366	1503	2749#			
LT34X	011476	2760	2764#			
LT34XX	011502	2765#				
LT35	011506	1504	2769#			
LT35A	011610	2778	2782#	2786		
LT351T	011522	1505	2771#	2780		
LT35X	011650	2785	2789#			
LT36	011660	1506	2794#			
LT361T	011674	1507	2796#	2799		
LT36X	011770	2805	2809#			
LT37	012000	1508	2814#			
LT37A	012054	2819	2823#	2832		
LT37B	012106	2828#	2831			
LT371T	012014	1509	2816#	2821		
LT37X	012140	2829	2835#			

LT4	003576	1454	1455	1870#	
LT4A	003714	1877	1891#	1929	1935
LT4B	003756	1896	1900#		
LT4C	003764	1902#			
LT4D	004116	1898	1926#	1937	
LT4ERG	004144	1931	1933#		
LT4ER1	004126	1899	1930#		
LT4ER2	004136	1901	1932#		
LT4G	003644	1880#	1927		
LT4GO	003634	1878#			
LT4X	004174	1875	1888	1925	1938#
LT40	012150	1510	2840#		
LT40IT	012164	1511	2840	2842#	
LT40X	012260	2851	2857#		
LT40XX	012264	2858#			
LT41	012270	1512	2862#		
LT41IT	012304	1513	2862	2864#	
LT41X	012500	2887	2891	2893#	
LT42	012532	1514	2908#		
LT42A	012634	2922#	2925		
LT42B	012650	2923	2926#		
LT42B1	012676	2927	2932#		
LT42B2	012716	2931	2936#		
LT42C	012762	2943#	2946		
LT42D	012776	2944	2947#		
LT42E	013026	2948	2953#		
LT42IT	012570	1515	2914	2915#	
LT42X	013046	2952	2956	2958#	
LT43	013062	1516	2964#		
LT43C	013172	2980#	2983		
LT43D	013206	2981	2984#		
LT43E	013240	2985	2990#		
LT43F	013272	2994	2997#		
LT43IT	013120	1517	2968	2970#	
LT43X	013312	2989	3000	3002#	
LT43XX	013322	2965	3004#		
LT44	013326	1518	3007#		
LT44A	013412	3017#	3020		
LT44A1	013424	3018	3021#		
LT44B	013434	3023#	3026		
LT44C	013450	3024	3027#		
LT44D	013476	3028	3032#		
LT44E	013500	3033#			
LT44F	013540	3034	3041#		
LT44IT	013354	1519	3010	3011#	
LT44X	013560	3040	3044	3046#	
LT44XX	013570	3048#			
LT45	013574	1520	3052#		
LT45A	013702	3068#	3071		
LT45B	013716	3069	3072#		
LT45D	013746	3073	3077#		
LT45E	013660	3062#	3065		
LT45E1	013672	3063	3066#		
LT45F	014000	3081	3084#		
LT45IT	013622	1521	3055	3056#	
LT45X	014024	3088	3090#		

MSG1	020642	1564	1569*	3910#			
MSG10	021564	1856	3996#				
MSG11	021610	1851	4000#				
MSG12	021635	3302	3464	4004#			
MSG13	021644	3306	3468	4006#			
MSG14	021653	1967	4008#				
MSG15	021670	1971	4011#				
MSG16	021706	3313	3474	4014#			
MSG18	021716	1996	4016#				
MSG19	021735	2020	3160	4019#			
MSG2	021045	1761	3933#				
MSG2A	021106	1759	3939#				
MSG20	021754	2047	4022#				
MSG21	022006	2074	4027#				
MSG22	022044	2080	4032#				
MSG23	022065	2086	4035#				
MSG24	022120	2111	4040#				
MSG25	022146	2115	4044#				
MSG25A	022175	2119	4048#				
MSG26	022222	2137	4052#				
MSG27	022251	2163	2185	2206	2231	4056#	
MSG3	021235	1791	3955#				
MSG30	022265	1920	4059#				
MSG31	022273	3416	4061#				
MSG32	022311	3412	4064#				
MSG33	022327	3419	4067#				
MSG34	022350	4070#					
MSG35	022365	4073#					
MSG36	022403	4076#					
MSG37	022421	4079#					
MSG4	021257	1820	1826	3959#			
MSG40	022437	3717	4082#				
MSG41	022443	1725	4083#				
MSG42	022462	3038	4086#				
MSG43	022477	3522	4089#				
MSG44	022603	1570	4101#				
MSG45	022625	1579	4105#				
MSG46	022647	3331	4109#				
MSG47	022753	3393	4121#				
MSG5	021275	1823	3962#				
MSG50	023005	2520	4126#				
MSG51	023024	3271	4129#				
MSG53	023053	2995	4133#				
MSG54	023065	3082	4135#				
MSG55	023076	2933	4137#				
MSG56	023173	1631	4148#				
MSG57	023220	1613	4152#				
MSG58	023377	1622	4172#				
MSG59	023422	3677	4176#				
MSG6	021315	3290	3965#				
MSG60	023424	3686	4177#				
MSG62	023426	1561	4178#				
MSG63	023474	3634	4185#				
MSG64	023522	1902	4189#				
MSG65	023540	1905	4192#				
MSG66	023545	1908	4193#				

CZTEACO TM03-TE16/TU77 CTL 1
CZTEAC.P11 14-MAY-79 09:02

MACY11 30A(1052) 14-MAY-79 09:06 PAGE 135
D 11
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0133

SCATCH	1194#	1274
SCHAIN	1194#	1549
SCHNMO	1194#	1675
SRESTO	1194#	3897
SSAVE	1194#	3887
.SACT1	1194#	1274
.SEOP	1194#	1732

. ABS. 031370 000

ERRORS DETECTED: 0

DSKZ:CZTEAC,DSKZ:CZTEAC.SEQ/CRF/SOL=CZTEAC.SML/ML,CZTEAC.P11
RUN-TIME: 16 29 4 SECONDS
RUN-TIME RATIO: 277/51=5.3
CORE USED: 11K (21 PAGES)