

TE16, TU77

TM03/TE16, TU77 CLT1
CZTEADO

AH-A792D-MC
FICHE 1 OF 1

OCT 1983
COPYRIGHT © 77-83
MADE IN USA



The main body of the document is a large, dark grid containing numerous small, illegible entries. The grid is organized into approximately 10 columns and 20 rows. Each cell in the grid appears to contain a small table or form with multiple columns of text, but the text is too faint to be read. The overall appearance is that of a microfiche or a high-density data storage format.

.REM %

IDENTIFICATION

PRODUCT CODE: AC-A791D-MC
PRODUCT NAME: CZTEADO TM03-TE16/TU77 CONTROL LOGIC TEST PART I
DATE CREATED: 11 - JULY - 1983
MAINTAINER: TAPE 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 RESPONSIBILILTY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (©) 1977,1983 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

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

1. ABSTRACT

THIS PROGRAM IS DESIGNED TO SEQUENTIALLY TEST ALL CONTROL LOGIC FUNCTIONALY OF THE TMO3. EACH TEST WILL ATTEMPT TO ISOLATE FAILURES TO THE MODULE LEVEL AND PROVIDE PRINTOUT INFORMATION WHICH WILL IDENTIFY THE FAILING MODULE. THE CONTROL LOGIC TESTS TEST ALL ERROR AND STATUS CONDITIONS AS WELL AS ADDRESSING PROTOCOL AND OPERATIONAL LOGIC SEQUENCES. THE LEVEL OF FAULT ISOLATION IS POSSIBLE BECAUSE OF TMO3 THE STRUCTURE AND ITS MAINTAINENCE MODES.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP-11 PROCESSOR
- B. 8K OF CORE
- C. CONSOLE TTY
- D. TMO3 MAGTAPE CONTROLLER
- E. MASSBUS CONTROLLER (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 TM03 DRIVE #0, TE16 SLAVE #0.

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

4.2 SAMPLE START AT 200

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

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

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

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

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

5. CONSOLE SWITCH SETTINGS

CONTROL:

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

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

SW15: 1=HALT ON ERROR
0=CONTINUE
SW14: 1=LOOP ON ERROR (SCOPE)
0=CONTINUE
SW13: 1=DO NOT PRINT ERRORS
0=PRINT ALL ERRORS
SW12: 1=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)
UNEXPTCED 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 TM03 AT THE ADDRESSES SPECIFIED BY THE OPERATOR. TEST OCCURS IMMEDIATELY AFTER DRIVE SELECTION.

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

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8909

CIRCUITS PRINT REFERENCES

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

LOGIC TEST #2: REGISTER ADDRESSING

PURPOSE: CHECK THE REGISTER SELECT LINES

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

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

CIRCUITS PRINT REFERENCE

C-LINES	(MB1,2,3),(MBI3),(MBI4),(MBI5)
RH REGISTER SELECT	(BCTA)
TM03 REGISTER SELECT	(MBI2)
MASSBUS REGISTER SELECT LINES	(MB1,2)
PARITY TREE	(MBI4)
CPAR,ILR BITS	(MBI11)

LOGIC TEST #3: CONTROL BUS

PURPOSE: VERIFY THAT ALL CONTROL LINES PROPERLY TRANSMIT

ONES AND ZEROS.

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

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

CIRCUITS

PRINT REFERENCE

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

LOGIC TEST #4: SLAVE ADDRESSING

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

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

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

LIKELY FAULTS LOCATIONS: M8905-YB,M8937,CABLE,M9001,M8910,M9001YA,M8933

CIRCUITS

PRINT REFERENCE

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

LOGIC TEST #5: MAINTENANCE REGISTER BITS

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

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

LIKELY FAULT LOCATIONS: M8905-YB

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

<u>CURCUITS</u>	<u>PRINT REFERENCE</u>
TM03 REGISTER SELECT	(MBI2)
TC FLIP-FLOPS, MULTIPLEXERS	(MR6)

LOGIC TEST #7: FRAME COUNT BIT TEST

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

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

LIKELY FAULT LOCATIONS: M8909

CIRCUITS

PRINT REFERENCE

TM03 REGISTER SELECT	(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

TM03 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

<u>CIRCUIT</u>	<u>PRINT REFERENCE</u>
FCS	MB18
SET ILF	MB17
SET NEF	MB17
GO BIT	MB15
GO BIT MULTIPLEXER	MR6
SET ILR	MB12

LOGIC TEST #12: DRIVE READY BIT

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

PRINT REF TCCM7

LOGIC TEST #13: INTERRUPT TEST

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

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

LIKELY FAULT LOCATION:

<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
MBI3

LOGIC TEST #16: STATUS AT EOT,ON LINE, LOADED, NO WRITE RING

PURPOSE: TO TEST EOT,SSC,SLA

PROGRAMMED SEQUENCE: THE OPERATOR IS INSTRUCTED TO MOVE TO EOT
AND PLACE THE DRIVE ON LINE. EOT,SSC,SLA ARE CHECKED IN
ADDITION TO ATA,MOL,WEL,DPR,DRY

LIKELY FAULT LOCATION: M8910,SLAVE CABLE,M8933

CIRCUIT

PRINT REFERENCE

SSC
EOT
SLA

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

LOGIC TEST #17: STATUS AT ONLINE LOADED

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

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

LOGIC TEST #20: ILLEGAL FUNCTION

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

LIKELY FAULT LOCATION: M8909

CIRCUIT

PRINT REFERENCE

SET ILF DECODE
ILF FLOP
ILF MULTIPLEXER

MB15,MB17
MB11
MB10

LOGIC TEST #21: REGISTER MODIFICATION REFUSED

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

LIKELY FAULT LOCATION: M8909

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

LOGIC TEST #22: CONTROL BUS PARITY (CPAR)

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

LIKELY FAULT LOCATIONS: M8909

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

LOGIC TEST #23: FORMAT ERROR (FMT)

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

LIKELY FAULT LOCATIONS: M8905-YB, M8906, M8909

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

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

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

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

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

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

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

LIKELY FAULT LOCATION: M8909

CIRCUIT

PRINT REFERENCE

NEF FLOP
NEF MULTIPLEXER
SET NEF

MBI11
MBI10
MBI7

LOGIC TEST #26: FRAME COUNT ERROR

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

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

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

CIRCUITS

PRINT REFERENCE

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

MB1
MB19
MBI11
TCCM5
MR5

LOGIC TEST #27: ILLEGAL REGISTER

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

LIKELY FAULT LOCATIONS: MASSBUSS, M8909

CIRCUITS

PRINT REFERENCE

REGISTER SELECT LINES
REGISTER SELECT DECODE
ILR FLOP

MB1, MB2
MBI2
MBI11

LOGIC TEST #30: DRIVE TIMING ERROR

PROGRAMMED SEQUENCE:

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

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

CIRCUITS

PRINT REFERENCES

DTE FLOP
CRIPPLE OCCUPIED FUNCTION
WRP 3 FUNCTION
PREVIOUS OCCUPIED CHECK
CHECK FOR WCLK
MM CLK

MBI11
MR5
MR5
MBI7
BF2
MR5

LOGIC TEST 31: OPERATION INCOMPLETE (OPI)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8933, M8909

CIRCUITS

PRINT REFERENCES

OPI TIMER
OPI FLOP
OPI TIMER CONTROL

TCCM5
MBI11
MBI7

LOGIC TEST 32: UNSAFE (UNS)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8909, M8910, SLAVE CABLE

CIRCUITS

PRINT REFERENCES

UNSAFE FLOP
SET UNSAFE
MOL GENERATION

MBI11
MBI7
LAW6

LOGIC TEST 33: POSITIONING IN PROGRESS (PIP)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8909, M8933

CIRCUITS

PRINT REFERENCES

SPACE FUNCTION DECODE
PIP GENERATION
STATUS REGISTER

MB15
TCCM7
TCCM7

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

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

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

CIRCUITS

PRINT REFERENCES

DENSITY BITS
DENSITY LINES
PES CIRCUIT
PES STATUS BIT

MR6
SBC
SC3
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 IS CHECKED FOR ONE. WAM 3 CODE IS LOADED, READ COMMAND IS ISSUED. AFTER A DELAY ACCL IS CHECKED FOR ZERO.

LIKELY FAULT LOCATIONS: M8933, 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
TAPE MARK MULTIPLEXER
ENVELOPE SIGNALS
WRITE DATA BUFFER
RDA MULTIPLEXERS
WRITE TAPE MARK FUNCTION
WAMO SIGNAL

TCPE4, TCPE5
TCCM7
DS 3, 5, 7
TCCM2
TCCM6
MBI5
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 TMO3 WRITE DATA LINES SHOULD BE ASSERTED.

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

LIKELY FAULT LOCATIONS: M8505, M8933, M8910, M8934

CIRCUITS

PRINT REFERENCES

MM FUNCTION DECODE
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
BIT STROBE CIRCUIT
DEAD TRACK FLOP
DEAD TRACK REGISTER

MR5
DS4
DS5, TCPE2
MR4

LOGIC TEST 45: PE INCORRECTABLE DATA

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

LIKELY FAULT LOCATIONS: M8932, M8901

CIRCUIT

PRINT REFERENCE

INC ERROR, PEF,

TCPE2

LOGIC TEST 46: PE FORMAT

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

LIKELY FAULT LOCATIONS: M8932, M8933, M8905-YB

CIRCUITS

PEF.
WRITE BUFFER
MM DECODE

PRINT REFERENCES

TCPE2
TCCM2
MR5

LOGIC TEST 47: FRAME COUNT OVERFLOW

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

LIKELY FAULT LOCATION: M8909

FRAME COUNT REGISTER MB18

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

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

PROGRAM SEQUENCE:

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

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

THIS TEST IS THE COMPLEMENT OF LOGIC TEST 50 ABOVE.

%


```
1211 .LIST BIN,LOC,SEQ
1212 .TITLE CZTEADO TM03-TE16/TU77 CTL I
1213 .CONTROL LOGIC TEST PART I
1214 .AC-A791D-MC
1215 .FEB 77
1216 .J.G. ADAMS
1217 .REVISED MAY 1978 BY J. G. ADAMS ;++B CHANGED MODULE REFERENCES TO
1218 ;++B REFLECT TM03 MODULES
1219 ;++B ADDED TU77 TEST CALABILITY
1220 .REVISED NOV 1978 BY M. PAGE ;+ INDICATES ENHANCEMENTS TO
1221 ; THE ORIGINAL REV (DZTEAA)
1222 ; NECESSARY FOR TM03
1223 .REVISED MAY,1983 BY B. LEBLANC ;BL FIXED AIDS #CC0001220
1224
1225 ;
1226 .MCALL .$ACT11,.$EOP,$CATCH,$SAVE,$RESTORE,$CHAIN,$CHNMODE
1227 .NLIST MC
1228 .LIST ME
1229 .ENABLE ABS,AMA
1230
1231
1232 ;CONSOLE SWITCHES*****
1233 ;
1234 ;SW15: 1=HALT ON ERROR
1235 ; 0=CONTINUE
1236 ;SW14: 1=LOOP ON ERROR
1237 ; 0=CONTINUE
1238 ;SW13: 1=DO NOT PRINT ERRORS
1239 ; 0=PRINT ERRORS
1240 ;SW12: 0=HALT AT END OF PASS
1241 ; 1=CONTINUOUS CYCLE
1242 ;SW11: 1=INHIBIT ITERATIONS
1243 ; 0=DO ITERATIONS
1244 ;SW10: 1=HALT AT END OF EACH TEST
1245 ; 0=CONTINUE
1246 ;SW9: 1=DO MANUAL INTERVENTION TESTS
1247 ; 0=INHIBIT MANUAL INTERVENTION
1248 ;SW0-5: SELECT TEST NUMBER :: 00=ALL TESTS
```



```

1297                                     ;REGISTER EQUIVS*****
1298
1299         000000         R0=%0
1300         000001         R1=%1
1301         000002         R2=%2
1302         000003         R3=%3
1303         000004         R4=%4
1304         000005         R5=%5
1305         000006         SP=%6
1306         000007         PC=%7
1307
1309
(1)                                     ;ACT11 HOOK *****
(1)         000764         $$VPC=          ;SAVE CURRENT LOCATION CTR
(1)         000042         .=42
(1) 000042 000000         .WORD 0
(1)         000046         .=46
(1) 000046 002664         .WORD SENDAD      ;SET LOCATION 46
(1)         000052         .=52
(1) 000052 000000         .WORD 0          ;SET LOCATION 52 = 0
(1)         000764         .=$$VPC        ;RESTORE LOCATION CTR
(1)
1310                                     ;TTY INTERRUPT VECTOR*****
1311
1312         000060         .=60
1313 000060 017254         .WORD TTINT    ;TTY INTERRUPT HEADER ADDRESS
1314 000062 000340         .WORD 340    ;PRIORITY LEVEL 7
1315
1316                                     ;SOFTWARE SWITCH REGISTER*****
1317                                     ;USED IF HARDWARE SWR = 177777 OR NOT AVAILABLE
1318         000176         .=176
1319 000176 000000         SWREG: .WORD 0          ;SOFTWARE SWITCH REGISTER
1320
1321                                     ;START ADDRESS*****
1322         000200         .=200
1323 000200 000137 001332         JMP START ;PROGRAM START
1324
1325                                     ;RESTART ADDRESS*****
1326         000210         .=210
1327 000210 000137 002224         JMP ST2
1328
1329                                     ;TM03 INTERRUPT VECTOR*****
1330
1331         000224         .=224
1332 000224 017244         MTINT          ;TAPE INTERRUPT HANDLER ADDRESS
1333 000226 000340
1334

```

```
1336
1337      000510
1338      .=510
1339      :MASS BUS REGISTER EQUIVS*****
1340 000510 172440      C1: 172440
1341 000512 172442      WC: 172442
1342 000514 172444      BA: 172444
1343 000516 172446      FC: 172446
1344 000520 172450      CS: 172450
1345 000522 172452      DS: 172452
1346 000524 172454      ER: 172454
1347 000526 172456      AS: 172456
1348 000530 172460      CC: 172460
1349 000532 172462      DB: 172462
1350 000534 172464      MR: 172464
1351 000536 172466      DT: 172466
1352 000540 172470      SN: 172470
1353 000542 172472      TC: 172472
1354
1355      :ILLEGAL FUNCTION CODES
1356
1357 000544 005405      ILFT: 5405
1358 000546 007415      7415
1359 000550 016423      16423
1360 000552 020437      20437
1361 000554 022443      22443
1362 000556 025447      25447
1363 000560 031455      31455
1364 000562 033465      33465
1365 000564 036473      36473
1366
1367      :CONSTANTS*****
1368
1369 000566 177776      PSW: 177776      :PROCESSOR STATUS
1370 000570 177570      SWR: 177570      :SWITCH REGISTER
1371 000572 177560      TKS: 177560      :TTY READER STATUS
1372 000574 177562      TKB: 177562      :TTY READ BUFFER
1373 000576 177564      TPS: 177564      :TTY PUNCH STATUS
1374 000600 177566      TPB: 177566      :TTY PUNCH BUFFER
1375 000602 000020      ITAMT: 20      :ITERATION AMOUNT
1376 000604 000224      VECT: 224      :INTERRUPT VECTOR(RH)
1377 000606 172440      REGS: 172440      :STARTING REGISTER ADDRESS
```



```
1379                                     :FLAGS AND COUNTERS*****
1380
1381 000610 000000 TOB: 0
1382 000612 000000 TIB: 0
1383 000614 000000 HDRFL: 0
1384 000616 000000 EMADDR: 0
1385 000620 000000 DRVN: 0
1386 000622 000000 TR00: 0
1387 000624 000000 TR01: 0
1388 000626 000000 TR02: 0
1389 000630 000000 TR03: 0
1390 000632 000000 TR04: 0
1391 000634 000000 TR05: 0
1392 000636 000000 TR06: 0
1393 000640 000000 TR07: 0
1394 000642 000000 TR10: 0
1395 000644 000000 TR11: 0
1396 000646 000000 TR12: 0
1397 000650 000000 TR13: 0
1398 000652 000000 TR14: 0
1399 000654 000000 TR15: 0
1400 000656 000000 NRZOF: 0
1401 000660 000000 SLVN: 0
1402 000662 000000 PFLG: 0
1403 000664 000000 RTRN: 0
1404 000666 000000 ERADD: 0
1405 000670 000000 TEMP1: 0
1406 000672 000000 TEMP2: 0
1407 000674 000000 TEMP3: 0
1408 000676 000000 ITCNT: 0
1409 000700 000000 SAV1: 0
1410 000702 000000 SAV2: 0
1411 000704 000000 SAV3: 0
1412 000706 000000 SCOLP: 0
1413 000710 000000 ITRLP: 0
1414 000712 000000 EXFL: 0
1415 000714 000000 ATAF: 0
1416 000716 000000 SLAF: 0
1417 000720 000000 SSCF: 0
1418 000722 000000 ERRF: 0
1419 000724 000000 ASF: 0
1420 000726 000000 SCF: 0
1421 000730 000000 TREF: 0
1422 000732 000000 PEXFL: 0
1423 000734 000000 STFLG: 0
1424 000736 000000 LTADD: 0
1425 000740 000000 T24FL: 0
1426 000742 000000 ADDFL: 0
1427 000744 000000 WAM: 0
1428 000746 000000 FUN: 0
1429 000750 000000 DATC: 0
1430 000752 000000 WTAD: 0
1431 000754 000000 DATAD: 0
1432 000756 000000 RDAD: 0
1433 000760 000000 W2FLG: 0
1434 000762 000000 DERFL: 0
```

1435 000764 000000
1436 000766 000000
1437 000770 000000
1438 000772 000000
1439 000774 000000
1440 000776 000000
1441 001000 000000
1442 001002 000000
1443 001004 000000
1444 001006 000000
1445 001010 000000
1446 001012 000000
1447 001014 000000
1448 001016 000003
1449 001020 000000
1450
1451
1452
1453 001022 000000
1454 001024 000000
1455 001026 000000
1456 001030 000000
1457
1458
1459
1460 001032 000005
1461 001034 000005
1462 001036 000012
1463 001040 000012
1464 001042 000000
1465 001044 000017
1466 001046 000017
1467 001050 000017
1468 001052 000017
1469 001054 000000

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

;++B INDICATES SLAVE TYPE (0/1 = TE16/TU77)
;PASS COUNTER
;+INDICATOR FOR NON-STANDARD CONFIG.
;+FLAG FOR NON-STANDARD CONFIG.

;EXPT WRAP STATUS*****

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

;CORE DUMP PATTERNS*****

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

1471
1472
1473
1474 001056 000000
1475 001060 000000
1476 001062 002734
1477 001064 002734
1478 001066 003210
1479 001070 003210
1480 001072 003414
1481 001074 003416
1482 001076 003576
1483 001100 003576
1484 001102 004200
1485 001104 004206
1486 001106 004370
1487 001110 004372
1488 001112 004504
1489 001114 004506
1490 001116 004620
1491 001120 004622
1492 001122 004744
1493 001124 004746
1494 001126 005170
1495 001130 005172
1496 001132 005364
1497 001134 005374
1498 001136 005466
1499 001140 005526
1500 001142 005576
1501 001144 005636
1502 001146 005706
1503 001150 005746
1504 001152 006020
1505 001154 006060
1506 001156 006132
1507 001160 006146
1508 001162 006274
1509 001164 006310
1510 001166 006440
1511 001170 006454
1512 001172 006564
1513 001174 006600
1514 001176 006714
1515 001200 006730
1516 001202 007240
1517 001204 007246
1518 001206 007372
1519 001210 007400
1520 001212 007606
1521 001214 007632
1522 001216 007724
1523 001220 007746
1524 001222 010242
1525 001224 010250
1526 001226 011072

:LOGIC TEST ENTRY TABLE*****

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

1527	001230	011106	LT32IT
1528	001232	011250	LT33
1529	001234	011264	LT33IT
1530	001236	011352	LT34
1531	001240	011366	LT34IT
1532	001242	011506	LT35
1533	001244	011522	LT35IT
1534	001246	011660	LT36
1535	001250	011674	LT36IT
1536	001252	012000	LT37
1537	001254	012014	LT37IT
1538	001256	012150	LT40
1539	001260	012164	LT40IT
1540	001262	012270	LT41
1541	001264	012304	LT41IT
1542	001266	012532	LT42
1543	001270	012570	LT42IT
1544	001272	013062	LT43
1545	001274	013120	LT43IT
1546	001276	013326	LT44
1547	001300	013354	LT44IT
1548	001302	013574	LT45
1549	001304	013622	LT45IT
1550	001306	014040	LT46
1551	001310	014066	LT46IT
1552	001312	014302	LT47
1553	001314	014316	LT47IT
1554	001316	014472	LT50
1555	001320	014506	LT50IT
1556	001322	014646	LT51
1557	001324	014662	LT51IT
1558	001326	002620	
1559	001330	000051	

TADX: .WORD TEND
TLAST: .WORD 51

;CONTAINS # OF TESTS


```

1561 .EVEN
1562 ;PROGRAM START AND HOUSEKEEPING*****
1563
1564 ;NOTE: PROGRAM STARTS HERE ON START AT 200
1565 START: MOV #500,SP ;SET STACK POINTER
1566 MOV @#4,-(SP) ;SAVE ERROR TRAP VECTOR
1567 MOV @#6,-(SP) ;AND VECTOR +2
1568 MOV #1$,@#4 ;SET NEW VECTOR
1569 CLR @#6 ;AND PSW
1570 CMP #-1,@SWR ;USE SOFTWARE SWITCH IF HARDWARE
1571 BEQ 2$ ;IS = 177777
1572 BR 3$ ;OTHERWISE USE HARDWARE SWR
1573 1$: CMP (SP)+,(SP)+ ;RESET STACK PTR
1574 2$: MOV #SWREG,SWR ;SET SOFTWARE SWITCH REGISTER
1575 3$: MOV (SP)+,@#6 ;RESTORE ERROR TRAP VECTORS
1576 MOV (SP)+,@#4
1577 CLR SKAT ;CLEAR SKIP ADDRESS TEST FLAG
1578 CLR (PC)+ ;CLEAR CHAIN INDICATOR
(1) CHNFLG: .WORD 0 ;CHAIN MODE INDICATOR
(1) ;1/0 = CHAIN/NOT CHAIN MODE
(1) ;BRANCH IF IN DUMP MODE
(1) 001422 005737 000042 TST @#42
(1) 001426 001407 BEQ 50$
(1) 001430 012737 000176 000570 MOV #SWREG,SWR ;:INVOKE SOFTWARE SWR
(1) 001436 005237 001420 INC CHNFLG ;:SET CHNFLG = CHAIN MODE
(1) 001442 000137 002244 JMP TSCD ;:GO TO CHAIN ADDRESS
(1) 001446 50$:
1579 001446 000240 SCHN: NOP
1580 001450 122737 000006 000041 4$: CMPB #6,@#41 ;BRANCH IF NOT LOADED VIA TMDP
1581 001456 001005 BNE 5$
1582 001460 012704 023426 MOV #MSG62,R4 ;ADVISE USER TO REMOVE TMDP FROM
1583 001464 004737 017724 JSR PC,TTOUT ;UNIT UNDER TEST
1584 001470 000000 HALT
1585 001472 012704 020642 5$: MOV #MSG1,R4
1586 001476 004737 017724 JSR PC,TTOUT ;PRINT TITLE
1587 001502 005737 001420 TST CHNFLG ;SEE IF IN CHAIN MODE
1588 001506 001402 BEQ 6$ ;IF NOT: BR
1589 001510 000137 002244 JMP TSCD ;ELSE GO TO START OF TESTS
1590 001514 112737 000043 020642 6$: MOVB #'#,MSG1 ;DO NOT PRINT TITLE ON RESTART
1591 001522 012704 022603 MOV #MSG44,R4
1592 001526 004737 017724 JSR PC,TTOUT ;REQUEST REGISTER ADDRESS
1593 001532 013703 000606 MOV REGS,R3
1594 001536 004737 020052 JSR PC,OCTP ;PRINT CURRENT ADDRESS
1595 001542 012705 000606 MOV #REGS,R5 ;SET ADDRESS SAVE LOC
1596 001546 012701 000007 MOV #7,R1 ;SET SIZE OF RESPONSE
1597 001552 012702 176400 MOV #176400,R2 ;SET UPPER LIMIT
1598 001556 012703 172300 MOV #172300,R3 ;SET LOWER LIMIT
1599 001562 004737 017402 JSR PC,TTR ;GO GET RESPONSE
1600 001566 012704 022625 MOV #MSG45,R4
1601 001572 004737 017724 JSR PC,TTOUT ;REQUEST VECTOR
1602 001576 013703 000604 MOV VECT,R3
1603 001602 004737 020052 JSR PC,OCTP ;PRINT CURRENT VECTOR
1604 001606 012705 000604 MOV #VECT,R5 ;SET ADDRESS SAVE LOC
1605 001612 012701 000004 MOV #4,R1 ;SET SIZE OF RESPONSE
1606 001616 012702 000224 MOV #224,R2 ;SET UPPER LIMIT
1607 001622 012703 000150 MOV #150,R3 ;SET LOWER LIMIT
1608 001626 004737 017402 JSR PC,TTR ;GO GET RESPONSE

```

1609	001632	013700	000604		MOV	VECT,R0	:GET VECTOR
1610	001636	012720	017244		MOV	#MTINT,(R0)+	:LOAD INTERRUPT ADDRESS IN VECTOR
1611	001642	012710	000340		MOV	#340,(R0)	:LOAD PRIORITY
1612	001646	013700	000606		MOV	REGS,R0	:GET START OF REGS
1613	001652	012701	000016		MOV	#16,R1	:SET NUMBER OF REGS
1614	001656	012702	000510		MOV	#C1,R2	:GET START OF TABLE
1615	001662	010022		ST0:	MOV	R0,(R2)+	:BUILD TABLE
1616	001664	062700	000002		ADD	#2,R0	:BUMP ADDRESS
1617	001670	005301			DEC	R1	:SEE IF DONE
1618	001672	001373			BNE	ST0	:IF NOT: BR
1619	001674	012702	000610		MOV	#TOB,R2	
1620	001700	012700	000077		MOV	#77,R0	
1621	001704	005022		ST1:	CLR	(R2)+	:CLEAR FLAGS + COUNTERS
1622	001706	005300			DEC	R0	
1623	001710	001375			BNE	ST1	
1624	001712	012704	023236		MOV	#MS57A,R4	:+REQUEST IF JUMPER IS IN NON-STANDARD MODE
1625	001716	004737	017724		JSR	PC,TTOUT	:+
1626	001722	012705	001016		MOV	#JUMPER,R5	:+
1627	001726	012703	000000		MOV	#0,R3	:+LIMIT RESPONSE
1628	001732	012701	000002		MOV	#2,R1	:+SET CHAR. NUMBER TO 1
1629	001736	012702	000004		MOV	#4,R2	:+SET RANGE 0 - 4
1630	001742	004737	017402		JSR	PC,TTR	:+GET RESPONSE
1631	001746	022737	000002	001016	CMP	#2,JUMPER	:TEST FOR NON-STANDARD JUMPER.
1632	001754	001002			BNE	1\$	
1633	001756	004737	016666		JSR	PC,NOST	:GO TO MODIFY SCHEDLAR
1634	001762	012704	023220	1\$:	MOV	#MSG57,R4	:REQUEST TMO3 DRIVE #
1635	001766	004737	017724		JSR	PC,TTOUT	
1636	001772	013703	000620		MOV	DRVN,R3	:GET CURRENT DRIVE #
1637	001776	004737	020052		JSR	PC,OCTP	:PRINT IT
1638	002002	012705	000620		MOV	#DRVN,R5	:TTR ROUTINE RETURNS USER VALUE TO (R5)
1639	002006	012701	000002		MOV	#2,R1	:LIMIT RESPONSE
1640	002012	012702	000007		MOV	#7,R2	:LIMIT RANGE TO 0-7
1641	002016	012703	000000		MOV	#0,R3	
1642	002022	004737	017402		JSR	PC,TTR	:GET USER RESPONSE
1643	002026	012704	023377		MOV	#MSG58,R4	:REQUEST TE16 SLAVE #
1644	002032	004737	017724		JSR	PC,TTOUT	
1645	002036	013703	000660		MOV	SLVN,R3	:GET CURRENT SLAVE #
1646	002042	004737	020052		JSR	PC,OCTP	:AND PRINT IT
1647	002046	012705	000660		MOV	#SLVN,R5	:TTR ROUTINE RETURNS RESPONSE TO (R5)
1648	002052	012701	000002		MOV	#2,R1	:LIMIT RESONSE TO 1 CHARACTER
1649	002056	012702	000007		MOV	#7,R2	:BETWEEN 0 AND 7
1650	002062	012703	000000		MOV	#0,R3	
1651	002066	004737	017402		JSR	PC,TTR	:GET USER RESPONSE
1652	002072	012704	023173		MOV	#MSG56,R4	
1653	002076	004737	017724		JSR	PC,TTOUT	:REQUEST STATIC ONLY
1654	002102	013703	001006		MOV	STATC,R3	:GET CURRENT VALUE
1655	002106	004737	020052		JSR	PC,OCTP	:AND TYPE IT
1656	002112	012705	001006		MOV	#STATC,R5	:SET ADDRESS OF STATIC FLAG
1657	002116	012701	000002		MOV	#2,R1	:SET SIZE OF RESPONSE
1658	002122	012702	000001		MOV	#1,R2	:SET UPPER LIMIT
1659	002126	012703	000000		MOV	#0,R3	:SET LOWER LIMIT
1660	002132	004737	017402		JSR	PC,TTR	:GET RESPONSE
1661							
1662	002136	012704	023552		MOV	#MSG67,R4	:++B REQUEST SLAVE TYPE 'TE16 OR TU77'
1663	002142	004737	017724		JSR	PC,TTOUT	:++B
1664	002146	013703	001010		MOV	SLVTYP,R3	:++B GET CURRENT SLAVE TYPE

1665	002152	004737	020052		JSR	PC,OCTP	;++B TYPE CURRENT VALUE
1666	002156	012705	001010		MOV	#SLVTYP,R5	;++B GET ADDRESS OF SLVTYP FLAG
1667	002162	012701	000002		MOV	#2,R1	;++B SET SIZE OF RESPONSE
1668	002166	012702	000001		MOV	#1,R2	;++B SET UPPER LIMIT
1669	002172	012703	000000		MOV	#0,R3	;++B SET LOWER LIMIT
1670	002176	004737	017402		JSR	PC,TTR	;++B GET RESPONSE
1671	002202	005737	001010		TST	SLVTYP	:IS IT A TU77
1672	002206	001406			BEQ	ST2	:BRANCH IF NOT
1673	002210	012737	116741	006016	MOV	#116741,STWD16	:SET UP TEST WORD FOR TEST 16
1674	002216	012737	110741	006130	MOV	#110741,STWD17	:SET UP TEST WORD FOR TEST 17
1675							
1676							
1677							
1678	002224	012706	000500				
1679	002230	005037	001002				
1680	002234	005037	001014				
1681	002240	004737	020500				

					:START 210			
					ST2:	MOV	#500,SP	:SET STACK PTR
						CLR	RDRVF	:CLEAR REVERSE FLAG
						CLR	PCNTR	:CLEAR PASS COUNTER
						JSR	PC,GTSWR	:GET SWITCHES

```

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



```

1716 002576 002410          BLT      TEND      ;IF SO: BR
1717 002600 006300          ASL      R0
1718 002602 006100          ROL      R0      ;SET TABLE MODIFIER
1719 002604 012737 001056 000736  MOV     #TSTTBL,LTADD
1720 002612 060037 000736  ADD     R0,LTADD  ;SET TEST POINTER
1721 002616 000730          BR       TSCD1
1722 002620 005737 001420  TEND:   TST      CHNFLG  ;BRANCH IF IN CHAIN MODE
1723 002624 001270          BNE     NXTSLV  ;STEP TO NEXT SLAVE
1724 002626 012704 022443  $DONE:  MOV     #MSG41,R4
1725 002632 004737 017724  JSR    PC,TTOUT ;PRINT END OF PASS
1726 002636 013703 001014  MOV    PCNTR,R3
1727 002642 004737 020052  JSR    PC,OCTP  ;PRINT PASS NUMBER
1728 002646 005000          CLR     R0      ;DELAY WAITING FOR
1729 002650 005300          1$:    DEC     R0
1730 002652 001376          BNE     1$
1731 002654 013700 000042  MOV     @#42,R0  ;GET ACT11 RETURN ADDRESS
(1) 002660 001405          BEQ    HERE     ;BRANCH IF NOT ACT11
(1) 002662 000005          RESET
(1) 002664 004710  $ENDAD: JSR    PC,(R0)
(1) 002666 000240          NOP
(1) 002670 000240          NOP
(1) 002672 000240          NOP
(1) 002674 000240          HERE:   NOP
1732 002676 005737 001420  TST     CHNFLG  ;BRANCH IF IN CHAIN MODE
1733 002702 001005          BNE     TENDX
1734 002704 032777 010000 175656  BIT     #10000,@SWR ;SEE IF HALT ON PASS
1735 002712 001401          BEQ    TENDX   ;IF NOT: BR
1736 002714 000000          HALT
1737 002716 012737 000001 001012  TENDX:  MOV     #1,SKAT ;SET SKIP ADDRESS TEST FLAG
1738 002724 005237 001014  INC    PCNTR   ;BUMP PASS COUNTER
1739 002730 000137 002244  JMP    TSCD   ;RESTART
1740          ;LOGIC TEST 1: DRIVE ADDRESSING*****
1741
1742 002734 012737 023672 000616  LT1:   MOV     #MSLT1,EMADDR ;++B SET ERROR MSG HDR ADDRESS
1743 002742 013737 000620 000674  MOV     DRVN,TEMP3 ;GET DRIVE # TO BE TESTED
1744 002750 013701 000620  MOV     DRVN,R1
1745 002754 005737 001012  TST     SKAT   ;SEE IF SKIP ADDRESS TESTS
1746 002760 001403          BEQ    1$     ;IF NOT: BR
1747 002762 005737 000734  TST     STFLG  ;SEE IF SINGLE TEST
1748 002766 001506          BEQ    LT1X   ;IF NOT: BR
1749 002770 032777 001000 175572  1$:    BIT     #1000,@SWR ;BRANCH IF MAN INTERVENTION
1750 002776 001430          BEQ    LT1A   ;NOT SELECTED
1751 003000 012704 021106  LT1G0: MOV     #MSG2A,R4
1752 003004 004737 017176  JSR    PC,INST ;PRINT TEST INSTRUCTIONS
1753 003010 012704 021045  LT1G:  MOV     #MSG2,R4
1754 003014 004737 017724  JSR    PC,TTOUT ;REQUEST DRIVE NUMBER
1755 003020 012705 000674  MOV     #TEMP3,R5 ;TTR ROUTINE RETURNS RESPONSE TO (R5)
1756 003024 012701 000002  MOV     #2,R1
1757 003030 012702 000007  MOV     #7,R2
1758 003034 012703 000000  MOV     #0,R3
1759 003040 004737 017402  JSR    PC,TTR  ;GET DRIVE NUMBER
1760 003044 005737 000670  TST     TEMP1  ;SEE IF ANOTHER DRIVE
1761 003050 001455          BEQ    LT1X   ;IF NOT: BR
1762 003052 005001          CLR     R1     ;SELECT DRIVE 0
1763 003054 012700 000010  MOV     #10,R0 ;SET NUMBER OF DRIVES
1764 003060 012777 000040 175432  LT1A:  MOV     #40,ACS ;INIT

```



```

1790                                     ;LOGIC TEST 2: REGISTER ADDRESSING*****
1791
1792 003210 000240                       LT2:  NOP
1793 003212 012777 000040 175300        LT2IT: MOV  #40,@CS          ;INIT
1794 003220 013777 000620 175272        MOV  DRVN,@CS        ;SELECT DRIVE
1795 003226 012737 023746 000616        MOV  #MSLT2,EMADDR  ;SAVE LT2 HEADER ADDRESS
1796 003234 012705 000510                MOV  #C1,R5          ;SET ADDRESS OF FIRST REGISTER
1797 003240 012700 000016                MOV  #16,R0          ;SET NUMBER OF REGISTERS
1798 003244 012702 000622                MOV  #TR00,R2        ;SET START OF REGISTER BUFFER
1799 003250 011501                       LT2A: MOV  (R5),R1
1800 003252 011112                       MOV  (R1),(R2)        ;READ REGISTER
1801 003254 032777 020000 175226        BIT  #20000,@C1      ;SEE IF ERROR
1802 003262 001402                       BEQ  LT2B             ;IF NOT: BR
1803 003264 004737 003314                JSR  PC,LT2ER1        ;ELSE GO TO ERROR 1
1804 003270 032777 000002 175226        LT2B: BIT  #2,@ER      ;SEE IF ILR
1805 003276 001402                       BEQ  LT2C             ;IF NOT: BR
1806 003300 004737 003332                JSR  PC,LT2ER2        ;ELSE GO TO ERROR 2
1807 003304 022225                       LT2C: CMP  (R2)+,(R5)+ ;BUMP ADDRESS
1808 003306 005300                       DEC  R0
1809 003310 001357                       BNE  LT2A             ;CONTINUE FOR ALL REGISTERS
1810 003312 000434                       BR   LT2X
1811
1812 003314 012737 000002 000712        LT2ER1: MOV #2,EXFL    ;FLAG NOT EXPECTED
1813 003322 012737 021257 000666        MOV  #MSG4,ERADD     ;POINT TO CONTROLLER ERROR
1814 003330 000415                       BR   LT2ERG          ;GO TO ERROR
1815 003332 012737 000002 000712        LT2ER2: MOV #2,EXFL    ;FLAG NOT EXPECTED
1816 003340 012737 021275 000666        MOV  #MSG5,ERADD     ;POINT TO DRIVE ERROR
1817 003346 000406                       BR   LT2ERG          ;GO TO ERROR
1818 003350 012737 000001 000712        LT2ER3: MOV #1,EXFL    ;FLAG EXPECTED
1819 003356 012737 021257 000666        MOV  #MSG4,ERADD     ;POINT TO DRIVE
1820 003364 012737 003400 000706        LT2ERG: MOV #LT2LP,SCOLP ;SET SCOPE ADDRESS
1821 003372 004737 015226                JSR  PC,LTGER        ;GO PRINT
1822 003376 000207                       RTS  PC              ;ELSE CONTINUE
1823 003400 005726                       LT2LP: TST (SP)+      ;RESET STACK
1824 003402 000722                       BR   LT2A            ;LOOP
1825 003404 004737 016576                LT2X: JSR PC,ITER     ;GO SEE IF ITERATIONS
1826 003410 000137 002526                JMP  TSCD2           ;RETURN TO SCHED
  
```

```

1828                                     ;LOGIC TEST 3: CONTROL BUS*****
1829
1830 003414 000240                       LT3:  NOP
1831 003416 012737 024025 000616 LT3IT: MOV  #MSLT3,EMADDR ;SET TEST HEADER
1832 003424 012701 000001                MOV  #1,R1 ;PRESET PATTERN 1
1833 003430 012700 000020                MOV  #20,R0 ;SET PATTERN CHANGE NUMBER
1834 003434 004737 016724                LT3A: JSR PC,INIT1 ;GO INIT
1835 003440 010177 175052                MOV  R1,@FC ;WRITE TO FC
1836 003444 032777 000010 175052        BIT  #10,@ER ;SEE IF CPAR (TM03)
1837 003452 001012                        BNE  LT3ER1 ;IF SO: BR
1838 003454 017702 175036                LT3B: MOV  @FC,R2 ;READ FC
1839 003460 032777 020000 175022        BIT  #20000,@C1 ;SEE IF MCPE (RH)
1840 003466 001017                        BNE  LT3ER2 ;IF SO: BR
1841 003470 005300                        LT3C: DEC  R0 ;SEE IF DONE PATTERN CHANGES
1842 003472 001426                        BEQ  LT3X ;IF SO: BR
1843 003474 006301                        ASL  R1 ;CHANGE PATTERN
1844 003476 000756                        BR   LT3A ;CONTINUE
1845 003500 012737 021610 000666 LT3ER1: MOV  #MSG11,ERADD ;SET ERROR CODE
1846 003506 012737 003434 000706        MOV  #LT3A,SCOLP ;SET SCOPE ADDRESS
1847 003514 017702 174776                MOV  @FC,R2 ;GET DATA
1848 003520 004737 016334                JSR  PC,LTGER1 ;GO DO ERROR
1849 003524 000753                        BR   LT3B
1850 003526 012737 021564 000666 LT3ER2: MOV  #MSG10,ERADD ;SET ERROR CODE
1851 003534 012737 003454 000706        MOV  #LT3B,SCOLP ;SET SCOPE ADDRESS
1852 003542 004737 016334                JSR  PC,LTGER1 ;GO DO ERROR
1853 003546 000750                        BR   LT3C
1854 003550 105701                        LT3X: TSTB R1 ;SEE IF DONE PATTERN 2
1855 003552 100405                        BMI  LT3XX ;IF SO: BR
1856 003554 012701 000401                MOV  #401,R1 ;SET PATTERN 2
1857 003560 012700 000010                MOV  #10,R0 ;SET PATTERN CHANGE NUMBER
1858 003564 000723                        BR   LT3A ;DO PATTERN 2
1859 003566 004737 016576                LT3XX: JSR PC,ITER ;GO SEE IF ITERATIONS
1860 003572 000137 002526                JMP  TSCD2 ;RETURN TO SCHEDULAR
  
```



```

1862
1863 ;LOGIC TEST 4: SLAVE ADDRESSING*****
1864
1865 003576 013737 000660 000674 LT4: MOV SLVN,TEMP3
1866 003604 013701 000660 MOV SLVN,R1
1867 003510 005737 001012 TST SKAT ;SEE IF SKIP ADDRESS TESTS
1868 003614 001403 BEQ 1$ ;IF NOT: BR
1869 003616 005737 000734 TST STFLG ;SEE IF SINGLE TEST
1870 003622 001564 BEQ LT4X ;IF NOT: BR
1871 003624 032777 001000 174736 1$: BIT #1000,@SWR ;BRANCH IF MAN INTERVETION
1872 003632 001430 BEQ LT4A ;NOT SELECTED
1873 003634 012704 021413 LT4G0: MOV #MSG8A,R4
1874 003640 004737 017176 JSR PC,INST ;PRINT TEST INSTRUCTIONS
1875 003644 012704 021352 LT4G: MOV #MSG8,R4
1876 003650 004737 017724 JSR PC,TTOUT ;REQUEST SLAVE
1877 003654 012705 000674 MOV #TEMP3,R5
1878 003660 012701 000002 MOV #2,R1
1879 003664 012702 000007 MOV #7,R2
1880 003670 012703 000000 MOV #0,R3
1881 003674 004737 017402 JSR PC,TTR ;GET SLAVE NUMBER
1882 003700 005737 000670 TST TEMP1 ;SEE IF SLAVE
1883 003704 001533 BEQ LT4X ;IF NOT: BR
1884 003706 005001 CLR R1 ;SELECT SLAVE 0
1885 003710 012700 000010 MOV #10,R0 ;SET NUMBER OF SLAVES
1886 003714 012777 000040 174576 LT4A: MOV #40,@CS ;INIT
1887 003722 013777 000620 174570 MOV DRVN,@CS ;SELECT DRIVE
1888 003730 010177 174606 MOV R1,@TC ;SELECT SLAVE
1889 003734 017703 174576 MOV @DT,R3 ;GET DT
1890 003740 020137 000674 CMP R1,TEMP3 ;SEE IF SHOULD HAVE SPR
1891 003744 001404 BEQ LT4B ;IF SO: BR
1892 003746 032703 002000 BIT #2000,R3 ;SEE IF SPR
1893 003752 001461 BEQ LT4D ;IF NOT: BR
1894 003754 000464 BR LT4ER1 ;GO TO ERROR 1
1895 003756 032703 002000 LT4B: BIT #2000,R3 ;SEE IF NO SLAVE PRESENT
1896 003762 001465 BEQ LT4ER2 ;(SPR=0)
1897 003764 012704 023522 LT4C: MOV #MSG64,R4 ;TYPE 'SLAVE TYPE = '
1898 003770 004737 017724 JSR PC,TTOUT
1899 003774 012702 000001 MOV #1,R2 ;PRESET SLAVE TYPE = TU77
1900 004000 012704 023540 MOV #MSG65,R4 ;SET UP TO TYPE 'TU77'
1901 004004 022777 142054 174524 CMP #142054,@DT ;BRANCH IF TU77
1902 004012 001412 BEQ 1$
1903 004014 012704 023545 MOV #MSG66,R4 ;CHANGE SLAVE TYPE TO 'TE16'
1904 004020 005302 DEC R2 ;CHANGE SLAVE TYPE TO TE16
1905 004022 022777 142051 174506 CMP #142051,@DT ;BRANCH IF TE16
1906 004030 001403 BEQ 1$
1907 004032 005302 DEC R2 ;CHENGE SLAVE TYPE TO ILLEGAL
1908 004034 012704 023662 MOV #MSG69,R4
1909 004040 004737 017724 1$: JSR PC,TTOUT ;TYPE SLAVE TYPE (TU77,TE16, OR ILLEGAL)
1910 004044 020237 001010 CMP R2,SLVTYP ;BRANCH IF HARDWARE SLAVE TYPE IS THE
1911 004050 001406 BEQ 4$ ;SAME AS USER SPACIFIED SLAVE TYPE
1912 004052 012704 023610 MOV #MSG68,R4 ;++B TYPE 'INCORRECT SLAVE TYPE'
1913 004056 004737 017724 JSR PC,TTOUT ;++B
1914 004062 000137 002620 JMP TEND ;++B EXIT TEST
1915 004066 012704 022265 4$: MOV #MSG30,R4
1916 004072 004737 017724 JSR PC,TTOUT ;PRINT SERIAL NUMBER TAG
1917 004076 017703 174436 MOV @SN,R3
  
```

1918	004102	004737	020376			JSR	PC,SNPT	:PRINT SERIAL NUMBER
1919	004106	032777	001000	174454		BIT	#1000,ASWR	:BRANCH IF NOT MANUAL INTERVENTION
1920	004114	001427				BEQ	LT4X	
1921	004116	005300			LT4D:	DEC	RO	
1922	004120	001651				BEQ	LT4G	:IF DONE ALL: BR
1923	004122	005201				INC	R1	:BUMP SLAVE
1924	004124	000673				BR	LT4A	:CONTINUE
1925	004126	012737	000001	000712	LT4ER1:	MOV	#1,EXFL	:FLAG EXPT: NOT RECEIVED
1926	004134	000403				BR	LT4ERG	
1927	004136	012737	000002	000712	LT4ER2:	MOV	#2,EXFL	:FLAG RECVD: NOT EXPT
1928	004144	012737	024112	000616	LT4ERG:	MOV	#MSLT4,EMADDR	:SET LT4 HEADER
1929	004152	012737	021542	000666		MOV	#MSG9,ERADD	:SET ERROR CONDITION
1930	004160	012737	003714	000706		MOV	#LT4A,SCOLP	:SET SCOPE ADDRESS
1931	004166	004737	015226			JSR	PC,LTGER	:GO TO ERROR
1932	004172	000751				BR	LT4D	:IF NO SCOPE: BR
1933	004174	000137	002526		LT4X:	JMP	TSCD2	:RETURN TO SCHED
1934								


```
1936                                     ;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST*****
1937
1938 004200 012737 024174 000616 LT5:  MOV #MSLT5,EMADDR ;SET TEST HEADER
1939 004206 004737 016724 LT5IT: JSR PC,INIT1 ;GO INIT
1940 004212 012700 000032      MOV #32,R0 ;SET LOOP FOR BITS 4-0
1941 004216 005001      CLR R1 ;SET TEST WORD
1942 004220 010177 174310      LT5A: MOV R1,@MR ;SEND TEST WORD TO MR
1943 004224 017702 174304      MOV @MR,R2 ;READ MR
1944 004230 042702 177740      BIC #177740,R2 ;MASK BITS 4-0
1945 004234 020102      CMP R1,R2 ;SEE IF EXPT = RECVD
1946 004236 001026      BNE LT5ER1
1947 004240 005300      LT5B: DEC R0
1948 004242 001402      BEQ LT5C ;IF DONE LOOP: BR
1949 004244 005201      INC R1 ;BUMP TEST WORD
1950 004246 000764      BR LT5A ;CONTINUE LOOP
1951 004250 012701 000015      LT5C: MOV #15,R1 ;SET TEST WORD + WAM 3
1952 004254 012700 001000      MOV #1000,R0 ;SET LOOP FOR BITS 15-7
1953 004260 010177 174250      LT5D: MOV R1,@MR ;LOAD MR
1954 004264 017702 174244      MOV @MR,R2 ;READ MR
1955 004270 042702 000140      BIC #140,R2 ;MASK OUT BITS 5,6
1956 004274 020102      CMP R1,R2 ;SEE IF EXPT = RECVD
1957 004276 001401      BEQ LT5E ;IF SO: BR
1958 004300 000416      BR LT5ER2 ;ELSE GO TO ERR 2
1959 004302 005300      LT5E: DEC R0
1960 004304 001425      BEQ LT5X ;IF DONE LOOP: BR
1961 004306 062701 000200      ADD #200,R1 ;BUMP TEST WORD
1962 004312 000762      BR LT5D ;CONTINUE LOOP
1963 004314 012737 021653 000666 LT5ER1: MOV #MSG14,ERADD ;SET ERROR CODE
1964 004322 012737 004220 000706      MOV #LT5A,SCOLP ;SET SCOPE ADDRESS
1965 004330 004737 016334      JSR PC,LTGER1 ;GO TO ERROR
1966 004334 000741      BR LT5B ;CONTINUE
1967 004336 012737 021670 000666 LT5ER2: MOV #MSG15,ERADD ;SET ERROR CODE
1968 004344 012737 004260 000706      MOV #LT5D,SCOLP ;SET SCOPE ADDRESS
1969 004352 004737 016334      JSR PC,LTGER1 ;GO TO ERROR
1970 004356 000751      BR LT5E ;CONTINUE
1971 004360 004737 016576      LT5X: JSR PC,ITER ;GO SEE IF ITERATIONS
1972 004364 000137 002526      JMP TSCD2 ;RETURN TO SCHED
1973
```



```
2001                                     ;LOGIC TEST 7: FRAME COUNT BIT TEST*****
2002
2003 004504 000240                       LT7:  NOP
2004 004506 012700 000003                 LT7IT: MOV #3,R0 ;SET TEST NUMBER
2005 004512 012737 024312 000616        LT7C:  MOV #MSLT7,EMADDR ;SET TEST HEADER
2006 004520 005001                       CLR R1 ;SET TEST WORD
2007 004522 004737 016724                 LT7A:  JSR PC,INIT1 ;GO INIT
2008 004526 010177 173764                 MOV R1,@FC ;CLEAR FRAME COUNT
2009 004532 017702 173760                 MOV @FC,R2 ;READ FC
2010 004536 020102                       CMP R1,R2 ;SEE IF EXPT = RECD
2011 004540 001010                       BNE LT7ER1
2012 004542 005300                       LT7B:  DEC R0 ;SEE IF DONE ALL
2013 004544 001417                       BEQ LT7X ;IF SO: BR
2014 004546 022700 000001                 CMP #1,R0 ;SEE IF RESET TEST
2015 004552 001757                       BEQ LT7C ;IF SO: BR
2016 004554 012701 177777                 MOV #-1,R1 ;SET TEST WORD TO -1
2017 004560 000760                       BR LT7A ;CONTINUE
2018 004562 012737 021735 000666        LT7ER1: MOV #MSG19,ERADD ;SET ERROR CODE
2019 004570 012737 004522 000706        MOV #LT7A,SCOLP ;SET SCOPE ADDRESS
2020 004576 004737 016334                 JSR PC,LTGER1 ;GO PRINT ERROR
2021 004602 000757                       BR LT7B ;ELSE CONTINUE
2022 004604 012700 000003                 LT7X:  MOV #3,R0 ;RESET TEST AMT
2023 004610 004737 016576                 JSR PC,ITER ;GO SEE IF ITERATIONS
2024 004614 000137 002526                 JMP TSCD2 ;RETURN TO SCHED
2025
```



```

2053
2054                                     ;LOGIC TEST 11: GO BIT SET RESET*****
2055
2056 004744 000240 LT11: NOP
2057 004746 012737 024437 000616 LT11IT: MOV #MSLT11,EMADDR ;SET TEST HEADER
2058 004754 004737 016724 JSR PC,INIT1 ;GO INIT
2059 004760 017702 173524 MOV @C1,R2 ;READ C1
2060 004764 032702 000001 BIT #1,R2 ;SEE IF GO=0
2061 004770 001030 BNE LT11E1
2062 004772 012777 000015 173534 LT11B: MOV #15,@MR ;SELECT WAM 3
2063 005000 005077 173512 CLR @FC ;ASSURE FCS = 1
2064 005004 052777 001700 173530 BIS #1700,@TC ;ASSURE FMT OK
2065 005012 012777 000071 173470 MOV #71,@C1 ;SET READ+GO
2066 005020 017702 173464 MOV @C1,R2 ;READ C1
2067 005024 032702 000001 BIT #1,R2 ;SEE IF GO =1
2068 005030 001424 BEQ LT11E2
2069 005032 004737 016724 LT11C: JSR PC,INIT1 ;GO INIT
2070 005036 017702 173446 MOV @C1,R2 ;READ C1
2071 005042 032702 000001 BIT #1,R2 ;SEE IF GO=0
2072 005046 001444 BEQ LT11X ;IF SO:BR
2073 005050 000430 BR LT11E3 ;ELSE GO TO ERROR 3
2074 005052 012737 022006 000666 LT11E1: MOV #MSG21,ERADD ;SET ERROR CODE
2075 005060 012702 000001 MOV #1,R2 ;SET REVD
2076 005064 005001 CLR R1 ;SET EXPT
2077 005066 012737 004746 000706 MOV #LT11IT,SCOLP ;SET SCOPE ADDRESS
2078 005074 004737 016334 JSR PC,LTGER1 ;GO PRINT ERROR
2079 005100 000734 BR LT11B ;ELSE CONTINUE
2080 005102 012737 022044 000666 LT11E2: MOV #MSG22,ERADD ;SET ERROR CODE
2081 005110 005002 CLR R2 ;SET RCVD
2082 005112 012701 000001 MOV #1,R1 ;SET EXPT
2083 005116 012737 004772 000706 MOV #LT11B,SCOLP ;SET SCOPE ADDRESS
2084 005124 004737 016334 JSR PC,LTGER1 ;GO PRINT ERROR
2085 005130 000740 BR LT11C ;ELSE CONTINUE
2086 005132 012737 022065 000666 LT11E3: MOV #MSG23,ERADD ;SET ERROR CODE
2087 005140 005001 CLR R1 ;SET EXPT
2088 005142 012702 000001 MOV #1,R2 ;SET RCVD
2089 005146 012737 005032 000706 MOV #LT11C,SCOLP ;SET SCOPE ADDRESS
2090 005154 004737 016334 JSR PC,LTGER1 ;GO PRINT ERROR
2091 005160 004737 016576 LT11X: JSR PC,ITER ;GO SEE IF ITERATIONS
2092 005164 000137 002526 JMP TSCD2 ;RETURN TO SCHED

```

```

2094
2095
2096
2097 005170 000240
2098 005172 012737 024504 000616
2099 005200 004737 016724
2100 005204 032777 000200 173310
2101 005212 001426
2102 005214 012777 000015 173312
2103 005222 005077 173270
2104 005226 052777 001700 173306
2105 005234 012777 000071 173246
2106 005242 032777 000200 173252
2107 005250 001020
2108 005252 004737 016724
2109 005256 032777 000200 173236
2110 005264 001033
2111 005266 000422
2112 005270 012737 022120 000666
2113 005276 012737 005172 000706
2114 005304 004737 016326
2115 005310 000741
2116 005312 012737 022146 000666
2117 005320 012737 005214 000706
2118 005326 004737 016326
2119 005332 000747
2120 005334 012737 022175 000666
2121 005342 012737 005252 000706
2122 005350 004737 016326
2123 005354 004737 016576
2124 005360 000137 002526

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



```
2126  
2127  
2128  
2129 005364 005000  
2130 005366 012737 024555 000616  
2131 005374 004737 016724  
2132 005400 012737 005456 000664  
2133 005406 005077 173076  
2134 005412 005077 173150  
2135 005416 052777 000100 173064  
2136 005424 005300  
2137 005426 001376  
2138 005430 012777 000340 173130  
2139 005436 012737 022222 000666  
2140 005444 012737 005374 000706  
2141 005452 004737 016326  
2142 005456 004737 016576  
2143 005462 000137 002526
```

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

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

```
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152 005466 032777 001000 173074 LT14: BIT #1000,@SWR ;SEE IF INHIB MAN TST  
2153 005474 001005 BNE LT14A ;IF NOT: BR  
2154 005476 005737 000734 TST STFLG ;SEE IF SINGLE TEST  
2155 005502 001433 BEQ LT14XX ;IF NOT: BR  
2156 005504 000137 016646 JMP INMT ;ELSE GO PRINT INHIB MSG  
2157 005510 012737 024622 000616 LT14A: MOV #MSLT14,EMADDR ;SET TEST HEADER  
2158 005516 012704 027027 MOV #MMSG1,R4 ;SET INSTRUCTION ONE  
2159 005522 004737 017176 JSR PC,INST ;GO DO INSTRUCTION  
2160 005526 004737 016724 LT14IT: JSR PC,INIT1 ;INIT, SELECT DRIVE + SLAVE  
2161 005532 012701 014602 MOV #14602,R1 ;SET TEST WORD  
2162 005536 017702 172760 MOV @DS,R2 ;ASSURE MOL,WRL,DPR,DRY,BOT  
2163 005542 020102 CMP R1,R2  
2164 005544 001410 BEQ LT14X ;IF SO: BR  
2165 005546 012737 005526 000706 MOV #LT14IT,SCOLP ;SET LOOP ADDRESS  
2166 005554 012737 022251 000666 MOV #MSG27,ERADD ;SET ERROR CODE  
2167 005562 004737 016334 JSR PC,LTGER1 ;GO PRINT ERROR  
2168 005566 004737 016576 LT14X: JSR PC,ITER ;GO SEE IF ITERATION  
2169 005572 000137 002526 LT14XX: JMP TSCD2 ;RETURN TO SCHED  
2170
```

: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*****


```
2172
2173
2174
2175 005576 032777 001000 172764 LT15: BIT #1000,@SWR :SEE IF INHIB MAN TST
2176 005604 001005 BNE LT15A :IF NOT: BR
2177 005606 005737 000734 TST STFLG :SEE IF SINGLE TEST
2178 005612 001433 BEQ LT15XX :IF NOT: BR
2179 005614 000137 016646 JMP INMT :ELSE GO PRINT INHIB MSG
2180 005620 012737 024731 000616 LT15A: MOV #MSLT15,EMADDR :SET TEST HEADER
2181 005626 012704 027125 MOV #MMSG2,R4
2182 005632 004737 017176 JSR PC,INST :PRINT INSTRUCTION
2183 005636 004737 016734 LT15IT: JSR PC,INIT2 :GO INIT, SELECT DRIVE, SLAV
2184 005642 012701 100700 MOV #100700,R1 :SET TEST WORD
2185 005646 017702 172650 MOV @DS,R2 :READ STATUS
2186 005652 020102 CMP R1,R2 :SEE OF EXPT=RCVD
2187 005654 001410 BEQ LT15X
2188 005656 012737 005636 000706 MOV #LT15IT,SCOLP :SET LOOP ADDRESS
2189 005664 012737 022251 000666 MOV #MSG27,ERADD :SET ERROR CODE
2190 005672 004737 016334 JSR PC,LTGER1 :GO PRINT ERROR
2191 005676 004737 016576 LT15X: JSR PC,ITER :GO SEE IF ITERATIONS
2192 005702 000137 002526 LT15XX: JMP TSCD2 :RETURN TO SCHED
```

```
2194  
2195  
2196  
2197 005706 032777 001000 172654 LT16: BIT #1000,@SWR ;SEE IF INHIB MAN TST  
2198 005714 001005 BNE LT16A ;IF NOT: BR  
2199 005716 005737 000734 TST STFLG ;SEE IF SINGLE TEST  
2200 005722 001433 BEQ LT16XX ;IF NOT: BR  
2201 005724 000137 016646 JMP INMT ;ELSE GO PRINT INHIB MSG  
2202 005730 012737 025021 000616 LT16A: MOV #MSLT16,EMADDR ;SET TEST HEADER  
2203 005736 012704 027146 MOV #MMSG3,R4  
2204 005742 004737 017176 JSR PC,INST ;GO PRINT INSTRUCTION  
2205 005746 004737 016734 LT16IT: JSR PC,INIT2 ;SELECT DRIVE,SLAVE  
2206 005752 013701 006016 MOV STWD16,R1 ;SET TEST WORD (ATA!MOL!WRL!EOT!DPR!DRY!SSC!SLA)  
2207 005756 017702 172540 MOV @DS,R2 ;READ STATUS  
2208 005762 020102 CMP R1,R2 ;SEE IF EXPT=RCVD  
2209 005764 001410 BEQ LT16X ;IF SO: BR  
2210 005766 012737 005746 000706 MOV #LT16IT,SCOLP ;SET LOOP ADDRESS  
2211 005774 012737 022251 000666 MOV #MSG27,ERADD ;SET ERROR CODE  
2212 006002 004737 016334 JSR PC,LTGER1 ;GO PRINT ERROR  
2213 006006 004737 016576 LT16X: JSR PC,ITER ;GO SEE IF ITERATION  
2214 006012 000137 002526 LT16XX: JMP TSCD2 ;RETURN TO SCHED  
2215 006016 116701 STWD16: .WORD 116701 ;SET UP TEST WORD FOR TE16  
2216 ;CONTENTS OF TEST WORD FOR  
2217 ;A TU77 IS 116741  
2218
```



```

2220
2221
2222
2223 006020 032777 001000 172542 LT17: BIT #1000,@SWR ;SEE IF INHIB MAN TST
2224 006026 001005 LT17A: BNE LT17A ;IF NOT: BR
2225 006030 005737 000734 TST STFLG ;SEE IF SINGLE TEST
2226 006034 001433 BEQ LT17XX ;IF NOT: BR
2227 006036 000137 016646 JMP INMT ;ELSE GO PRINT INHIB MSG
2228 006042 012737 025110 000616 LT17A: MOV #MSLT17,EMADDR ;SET TEST HEADER
2229 006050 012704 027204 MOV #MMSG4,R4
2230 006054 004737 017176 JSR PC,INST ;GO PRINT INSTRUCTION
2231 006060 004737 016734 LT17IT: JSR PC,INIT2 ;SELECT DRIVE, SLAVE
2232 006064 013701 006130 MOV STWD17,R1 ;SET TEST WORD
2233 006070 017702 172426 MOV @DS,R2 ;READ STATUS
2234 006074 020102 CMP R1,R2 ;SEE IF EXPT=RCVD
2235 006076 001410 BEQ LT17X ;IF SO: BR
2236 006100 012737 006060 000706 MOV #LT17IT,SCOLP ;SET LOOP ADDRESS
2237 006106 012737 022251 000666 MOV #MSG27,ERADD ;SET ERROR CODE
2238 006114 004737 016334 JSR PC,LTGER1 ;YES PRINT ERROR
2239 006120 004737 016576 LT17X: JSR PC,ITER ;GO SEE IF ITERATIONS
2240 006124 000137 002526 LT17XX: JMP TSCD2 ;RETURN TO SCHED
2241 006130 110701 STWD17: .WORD 110701 ;TEST WORD FOR TE16
2242 ;CONTENTS OF TEST WORD
2243 ;IS 110741 FOR A TU77
  
```

```
2245 :THE FOLLOWING 11 TESTS WILL TEST ALL POSSIBLE ERROR BITS
2246 :BY FORCING THEIR CONDITIONS THROUGH VARIOUS ILLEGAL PROGRAMMING
2247 :SEQUENCES AND USING THE MAINTENANCE WILL MODES AVAILABLE WITH TMO3
2248 :FOR EACH ERROR CONDITION SET THE APPROPRIATE STATUS WILL BE
2249 :CHECKED. IE: ERR, ATA, SLA, SC ETC.
2250
2251 :LOGIC TEST 20: ILLEGAL FUNCTION (ILF)*****
2252
2253 006132 012737 025171 000616 LT20: MOV #MSLT20,EMADDR :SET TEST HEADER
2254 006140 012737 006160 000706 MOV #LT20A,SCOLP :SET LOOP ADDRESS
2255 006146 012700 000022 LT20IT: MOV #22,R0 :SET NUMBER OF ILL CODES
2256 006152 012737 000544 000670 MOV #ILFT,TEMP1 :POINT TO START IF TABLE
2257 006160 004737 016724 LT20A: JSR PC,INIT1 :GO INIT, SELECT SLAVE + DRIVE
2258 006164 012777 177777 172320 MOV #-1,@WC :SET WC= -1
2259 006172 012701 000001 MOV #1,R1 :SET TEST WORD
2260 006176 117777 172466 172304 MOVB @TEMP1,@C1 :SET ILL CODE
2261 006204 017702 172314 MOV @ER,R2 :READ ER
2262 006210 030102 BIT R1,R2 :SEE IF EXPT=RCVD
2263 006212 001011 BNE LT20B :IF SO: BR
2264 006214 012737 027507 000666 MOV #TMS17,ERADD :SET ERROR CODE
2265 006222 012737 000001 000712 MOV #1,EXFL :SET EXPT FLG
2266 006230 004737 015220 JSR PC,LTGER0 :GO PRINT ERROR
2267 006234 000404 BR LT20C
2268 006236 020102 LT20B: CMP R1,R2 :SEE UNEXPECTED ERRORS
2269 006240 001402 BEQ LT20C :IF NOT: BR
2270 006242 004737 015206 JSR PC,LTGER3 :ELSE PRINT ERROR
2271 006246 005300 LT20C: DEC R0 :SEE IF DONE ALL ILL CODES
2272 006250 001403 BEQ LT20X :IF SO: BR
2273 006252 005237 000670 INC TEMP1 :BUMP ADDRESS
2274 006256 000740 BR LT20A :CONTINUE
2275 006260 004737 016576 LT20X: JSR PC,ITER :GO SEE IF ITERATION
2276 006264 004737 015666 JSR PC,DRVCLR
2277 006270 000137 002526 JMP TSCD2 :RETURN TO SCHED
```



```

2279
2280 ;LOGIC TEST 21: REGISTER MODIFICATION REFUSED(RMR)*****
2281
2282 006274 012737 025250 000616 LT21: MOV #MSLT21,EMADDR ;SET TEST HEADER
2283 006302 012737 006310 000706 MOV #LT21IT,SCOLP ;SET SCOPE LOOP ADDRESS
2284 006310 004737 016724 LT21IT: JSR PC,INIT1 ;GO INIT, SELECT SLAVE, DRIVE
2285 006314 052777 000300 172220 BIS #300,@TC ;SET FORMAT
2286 006322 012777 000015 172204 MOV #15,@MR ;SET WAM3
2287 006330 012777 000071 172152 MOV #71,@C1 ;SET READ+GO
2288 006336 005077 172154 CLR @FC ;ATTEMPT WRITE TO FC
2289 006342 012701 000004 MOV #4,R1 ;SET TEST WORD
2290 006346 017702 172152 MOV @ER,R2 ;GET ER
2291 006352 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2292 006354 001011 BNE LT21A ;IF SO: BR
2293 006356 012737 027523 000666 MOV #TMS19,ERADD ;SET ERROR CODE
2294 006364 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2295 006372 004737 015220 JSR PC,LTGER0 ;GO PRINT ERROR
2296 006376 000404 BR LT21B
2297 006400 020102 LT21A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2298 006402 001402 BEQ LT21B ;IF NOT: BR
2299 006404 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2300 ;++B LT21B: JSR PC,ITER ;++B DELETED GO SEE IF ITERATION
2301 006410 012703 040000 LT21B: MOV #40000,R3
2302 006414 005303 LT21XA: DEC R3 ;DELAY FOR ALPHA
2303 006416 001376 BNE LT21XA
2304 006420 004737 015052 JSR PC,EORPA ;GO DO EOR CLEAR
2305 006424 004737 015666 JSR PC,DRVCLR
2306 006430 004737 016576 JSR PC,ITER ;++B GO SEE IF ITERATION
2307 006434 000137 002526 JMP TSCD2 ;RETURN TO SCHED
  
```

```
2309
2310 ;LOGIC TEST 22: CONTROL BUS PARITY (CPAR)*****
2311
2312 006440 012737 025304 000616 LT22: MOV #MSLT22,EMADDR ;SET TEST HEADER
2313 006446 012737 006454 000706 MOV #LT22IT,SCLP ;SET SCOPE LOOP ADDRESS
2314 006454 004737 016724 LT22IT: JSR PC,INIT1 ;INIT, SELECT SLAVE+DRIVE
2315 006460 052777 000020 172032 BIS #20,@CS ;ENABLE EVEN PARITY ON MB
2316 006466 012777 177777 172022 MOV #-1,@FC ;WRITE TO FC
2317 006474 012701 000010 MOV #10,R1 ;SET TEST WORD
2318 006500 042777 000020 172012 BIC #20,@CS ;RESET PARITY TO ODD
2319 006506 017702 172012 MOV @ER,R2 ;GET ER
2320 006512 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2321 006514 001011 BNE LT22A ;IF SO: BR
2322 006516 012737 027531 000666 MOV #TMS20,ERADD ;SET ERROR CODE
2323 006524 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2324 006532 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2325 006536 000404 BR LT22X
2326 006540 020102 LT22A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2327 006542 001402 BEQ LT22X ;IF NOT: BR
2328 006544 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2329 006550 004737 016576 LT22X: JSR PC,ITER ;GO SEE IF ITERATION
2330 006554 004737 015666 JSR PC,DRVCLR
2331 006560 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```



```
2333  
2334  
2335  
2336 006564 012737 025341 000616 LT23: MOV #MSLT23,EMADDR ;SET TEST HEADER  
2337 006572 012737 006600 000706 MOV #LT23IT,SCOLP ;SET SCOPE ADDRESS  
2338 006600 004737 016724 LT23IT: JSR PC,INIT1 ;GO INIT SELECT DRIVE+SLAVE  
2339 :++B BIC #360,@TC ;++B DELETED SET ILLEGAL FORMAT  
2340 006604 052777 000360 171730 BIS #360,@TC ;++B SET ILLEGAL FORMAT FOR BOTH M8906 & M8915  
2341 006612 012701 000020 MOV #20,R1 ;SET TEST WORD  
2342 006616 012777 000015 171710 MOV #15,@MR ;SET WAM 3  
2343 006624 012777 000071 171656 MOV #71,@C1 ;SET READ+GO  
2344 006632 017702 171666 MOV @ER,R2 ;READ ER  
2345 006636 030102 BIT R1,R2 ;SEE IF EXPT=RCVD  
2346 006640 001011 BNE LT23A ;IF SO: BR  
2347 006642 012737 027540 000666 MOV #TMS21,ERADD ;SET ERROR CODE  
2348 006650 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG  
2349 006656 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR  
2350 006662 000404 BR LT23X  
2351 006664 020102 LT23A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS  
2352 006666 001402 BEQ LT23X ;IF NOT: BR  
2353 006670 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR  
2354 006674 004737 016576 LT23X: JSR PC,ITER ;GO SEE IF ITERATION  
2355 006700 004737 015052 JSR PC,EORPA  
2356 006704 004737 015666 JSR PC,DRVCLR  
2357 006710 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```

2359                                     ;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)*****
2360
2361 006714 012737 025406 000616 LT24:  MOV    #MSLT24,EMADDR ;SET TEST HEADER
2362 006722 012737 006730 000706      MOV    #LT24IT,SCOLP ;SET SCOPE ADDRESS
2363 006730 012737 000005 000602 LT24IT: MOV    #5,ITAMT
2364 006736 004737 016752      JSR    PC,INIT3      ;GO INIT, SELECT DRIVE+SLAVE
2365 006742 052777 000300 171572      BIS    #300,@TC      ;SET NORMAL FORMAT
2366 006750 012777 030040 171536      MOV    #WDATA,@BA    ;SET BA
2367 006756 012777 177760 171532      MOV    #-20,@FC      ;SET FC
2368 006764 012777 177770 171520      MOV    #-10,@WC      ;SET WC
2369 006772 012777 000013 171534      MOV    #13,@MR       ;SELECT WAM 2
2370 007000 012777 000061 171502      MOV    #61,@C1       ;SET WRITE+GO
2371 007006 052777 000020 171504      BIS    #20,@CS       ;FORCE EVEN PARITY
2372 007014 012701 000040      MOV    #40,R1        ;SET TEST WORD
2373 007020 012703 000004      MOV    #4,R3
2374 007024 005000      CLR    R0
2375 007026 005300      1$:   DEC    R0
2376 007030 001376      BNE    1$             ;DELAY
2377 007032 005303      DEC    R3
2378 007034 001374      BNE    1$
2379 007036 012700 000004      MOV    #4,R0
2380 007042 012777 000013 171464 LT24B: MOV    #13,@MR       ;CLOCK MR 4 TIMES
2381 007050 005300      DEC    R0
2382 007052 022700 000002      CMP    #2,R0         ;SEE IF DONE 1 BYTE
2383 007056 001002      BNE    LT24B0        ;IF NOT: BR
2384 007060 017701 171450      MOV    @MR,R1        ;ELSE GET BYTE 1
2385 007064 005700      LT24B0: TST    R0         ;SEE IF BYTE 2
2386 007066 001365      BNE    LT24B         ;IF NOT: BR
2387 007070 017704 171440      MOV    @MR,R4        ;GET BYTE 2
2388 007074 005000      CLR    R0
2389 007076 005300      LT24C: DEC    R0
2390 007100 001376      BNE    LT24C         ;DELAY
2391 007102 032777 000040 171414      BIT    #40,@ER       ;SEE IF DPAR IS SET
2392 007110 001023      BNE    LT24D         ;IF SO: BR
2393 007112 000301      SWAB   R1
2394 007114 042701 177400      BIC    #177400,R1    ;GET LOW BYTE
2395 007120 042704 000377      BIC    #377,R4
2396 007124 050401      BIS    R4,R1         ;GET HIGH BYTE
2397 007126 005237 000740      INC    T24FL        ;SET T24 FLAG
2398 007132 012737 027546 000666      MOV    #TMS22,ERADD ;SET ERROR CODE
2399 007140 012737 000001 000712      MOV    #1,EXFL      ;SET EXPT FLG
2400 007146 004737 015220      JSR    PC,LTGERO    ;GO PRINT ERROR
2401 007152 005037 000740      CLR    T24FL        ;CLEAR FLAG
2402 007156 000412      BR     LT24X
2403 007160 012701 000050      LT24D: MOV    #50,R1
2404 007164 017702 171334      MOV    @ER,R2        ;GET ERROR REGISTER
2405 007170 042702 020000      BIC    #20000,R2    ;MASK OPI
2406 007174 020102      CMP    R1,R2        ;SEE IF UNEXPECTED ERRORS
2407 007176 001402      BEQ    LT24X        ;IF NOT: BR
2408 007200 004737 015206      JSR    PC,LTGER3    ;ELSE GO PRINT ERROR
2409 007204 042777 000020 171306 LT24X: BIC    #20,@CS       ;RESET EVEN PARITY
2410 007212 004737 015052      JSR    PC,EORPA     ;GO DO EOR CLEAR
2411 007216 004737 015666      JSR    PC,DRVCLR    ;GO SEE IF DRIVE CLEAR OK
2412 007222 004737 016576      JSR    PC,ITER      ;GO SEE IF ITERATION
2413 007226 012737 000020 000602      MOV    #20,ITAMT
2414 007234 000137 002526      JMP    TSCD2        ;RETURN TO SCHED

```



```
2416  
2417  
2418  
2419 007240 012737 025446 000616 LT25: MOV #MSLT25,EMADDR ;SET TEST HEADER  
2420 007246 004737 016752 LT25IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE  
2421 007252 052777 000300 171262 BIS #300,@TC ;SET NORMAL FORMAT  
2422 007260 012777 177777 171230 MOV #-1,@FC ;SET ITLLEGAL FC  
2423 007266 012777 000013 171240 MOV #13,@MR ;SET WAM 2  
2424 007274 012777 000061 171206 MOV #61,@C1 ;LOAD WRITE+GO  
2425 007302 012701 004000 MOV #4000,R1 ;SET TEST WORD  
2426 007306 017702 171212 MOV @ER,R2 ;GET ER  
2427 007312 030102 BIT R1,R2 ;SEE IF EXPT=RCVD  
2428 007314 001014 BNE LT25A ;IF SO: BR  
2429 007316 012737 007246 000706 MOV #LT25IT,SCOLP ;SET LOOP ADDRESS  
2430 007324 012737 027634 000666 MOV #TMS31,ERADD ;SET ERROR CODE  
2431 007332 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLAG  
2432 007340 004737 015220 JSR PC,LTGER0 ;GO PRINT ERROR  
2433 007344 000404 BR LT25X  
2434 007346 020102 LT25A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS  
2435 007350 001402 BEQ LT25X ;IF NOT: BR  
2436 007352 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR  
2437 007356 004737 016576 LT25X: JSR PC,ITER ;GO SEE IF ITERATION  
2438 007362 004737 015666 JSR PC,DRVCLR  
2439 007366 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```

2441
2442 ;LOGIC TEST 26: FRAME COUNT ERROR(FCE)*****
2443
2444 007372 012737 025502 000616 LT26: MOV #MSLT26,EMADDR ;SET TEST HEADER
2445 007400 004737 016752 LT26IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2446 007404 005000 CLR R0
2447 007406 005300 1$: DEC R0
2448 007410 001376 BNE 1$ ;AWAIT OPI RESET
2449 007412 052777 000300 171122 BIS #300,@TC ;SET NORMAL FORMAT
2450 007420 012777 177770 171064 MOV #-10,@WC ;SET WC=-10
2451 007426 012777 177760 171062 MOV #-20,@FC ;SET FC=-20
2452 007434 012777 000013 171072 MOV #13,@MR ;SET WAM 3
2453 007442 012777 000061 171040 MOV #61,@C1 ;LOAD WRITE+GO
2454 007450 012701 001000 MOV #1000,R1 ;SET TEST WORD
2455 007454 005000 CLR R0
2456 007456 005300 2$: DEC R0
2457 007460 001376 BNE 2$ ;DELAY
2458 007462 012777 000025 171044 MOV #25,@MR ;LOAD MM EOR CLEAR
2459 007470 105077 171040 CLR @MR ;RESET MR
2460 007474 012703 000004 MOV #4,R3
2461 007500 005000 CLR R0
2462 007502 032777 001000 171014 3$: BIT #1000,@ER ;SEE IF FCE SET
2463 007510 001022 BNE 4$ ;IF SO: BR
2464 007512 005300 DEC R0
2465 007514 001372 BNE 3$ ;DELAY
2466 007516 005303 DEC R3
2467 007520 001370 BNE 3$
2468 007522 017702 170776 MOV @ER,R2 ;GET ER
2469 007526 012737 007400 000706 MOV #LT26IT,SCOLP ;SET SCOPE ADDRESS
2470 007534 012737 027613 000666 MOV #TMS28,ERADD
2471 007542 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2472 007550 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2473 007554 000406 BR LT26X
2474 007556 017702 170742 4$: MOV @ER,R2 ;GET ERROR REGISTER
2475 007562 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2476 007564 001402 BEQ LT26X ;IF NOT: BR
2477 007566 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2478 007572 004737 016576 LT26X: JSR PC,ITER ;GO SEE IF ITERATION
2479 007576 004737 015666 JSR PC,DRVCLR
2480 007602 000137 002526 JMP TSCD2 ;RETURN TO SCHED
  
```



```
2482  
2483 ;LOGIC TEST 27: ILLEGAL REGISTER(ILR)*****  
2484  
2485 007606 022737 172400 000510 LT27: CMP #172400,C1 ;SEE IF ADDRESSES OPEN  
2486 007614 001041 BNE LT27XX ;IF NOT: BR  
2487 007616 012737 007642 000706 MOV #LT27A,SCOLP ;SET SCOPE ADDRESS  
2488 007624 012737 025536 000616 MOV #MSLT27,EMADDR ;SET TEST HEADER  
2489 007632 012700 000020 LT27IT: MOV #20,R0 ;SET NUMBER OF ILR TESTS  
2490 007636 012701 172434 MOV #172434,R1 ;SET FIRST ILR ADDRESS  
2491 007642 004737 016752 LT27A: JSR PC,INIT3 ;GO INIT, SELECT DRIVE+SLAVE  
2492 007646 011103 MOV (R1),R3 ;ATTEMPT ILR READ  
2493 007650 032777 000002 170646 BIT #2,@ER ;SEE IF ILR=1  
2494 007656 001010 BNE LT27B ;IF SO: BR  
2495 007660 012737 000001 000712 MOV #1,EXFL ;SET EXPT-NOT RCVD FLAG  
2496 007666 012737 027435 000666 MOV #TMS10,ERADD ;SET ERROR CODE  
2497 007674 004737 015226 JSR PC,LTGER ;GO PRINT ERROR  
2498 007700 005300 LT27B: DEC R0 ;SEE IF DONE ALL  
2499 007702 001402 BEQ LT27X ;IF SO: BR  
2500 007704 005721 TST (R1)+ ;BUMP ADDRESS  
2501 007706 000755 BR LT27A ;CONTINUE TESTS  
2502 007710 004737 016576 LT27X: JSR PC,ITER ;GO SEE IF ITERATIONS  
2503 007714 004737 015666 JSR PC,DRVCLR  
2504 007720 000137 002526 LT27XX: JMP TSCD2 ;RETURN TO SCHED
```

```

2506
2507
2508
2509 007724 012737 027642 000666 LT30: MOV #TMS32,ERADD ;SET ERROR CODE
2510 007732 012737 025572 000616 MOV #MSLT30,EMADDR ;SET TEST HEADER
2511 007740 012737 007746 000706 MOV #LT30IT,SCOLP ;SET SCOPE ADDRESS
2512 007746 004737 016752 LT30IT: JSR PC,INIT3 ;INIT, SELECT DRIVE + SLAVE
2513 007752 052777 000300 170562 BIS #300,@TC ;SET NORMAL FORMAT
2514 007760 012701 010000 MOV #10000,R1 ;SET TEST WORD
2515 007764 012777 000017 170542 MOV #17,@MR ;CRIPPLE OCCUPIED
2516 007772 005077 170520 CLR @FC ;SET FC3
2517 007776 012777 000061 170504 MOV #61,@C1 ;LOAD WRITE+GO
2518 010004 032777 010000 170512 BIT #10000,@ER ;SEE IF DTE SET
2519 010012 001005 BNE LT30A ;IF SO: BR
2520 010014 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2521 010022 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2522 010026 004737 016752 LT30A: JSR PC,INIT3 ;GO INIT SELECT DRIVE,SLAVE
2523 010032 052777 000300 170502 BIS #300,@TC ;SET FORMAT
2524 010040 012701 010000 MOV #10000,R1 ;SET TEST WORD
2525 010044 005077 170446 CLR @FC ;SET FCS
2526 010050 012777 000015 170456 MOV #15,@MR ;SET WRAP 3
2527 010056 012777 000061 170424 MOV #61,@C1 ;LOAD WRITE+GO
2528 010064 012704 040000 MOV #40000,R4
2529 010070 005777 170446 LT30B: TST @TC ;SEE IF ALPHA
2530 010074 100015 BPL LT30C ;AWAIT ALPHA
2531 010076 005300 DEC R0
2532 010100 001373 BNE LT30B
2533 010102 013704 000616 MOV EMADDR,R4
2534 010106 004737 017724 JSR PC,TTOUT ;PRINT HEADER
2535 010112 012704 023005 MOV #MSG50,R4
2536 010116 004737 017724 JSR PC,TTOUT ;PRINT ALPHA ERROR
2537 010122 004737 016546 JSR PC,SCOPE
2538 010126 000435 BR LT30X
2539 010130 012777 000015 170376 LT30C: MOV #15,@MR ;CLOCK MR
2540 010136 012777 000015 170370 MOV #15,@MR ;CLOCK MR
2541 010144 005000 CLR R0
2542 010146 005300 LT30D: DEC R0
2543 010150 001376 BNE LT30D ;DELAY
2544 010152 032777 010000 170344 BIT #10000,@ER ;SEE IF DTE SET
2545 010160 001006 BNE LT30E ;IF SO: BR
2546 010162 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLG
2547 010170 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2548 010174 000412 BR LT30X
2549 010176 012701 010000 LT30E: MOV #10000,R1 ;SET TEST WORD
2550 010202 017702 170316 MOV @ER,R2 ;GET ERROR REGISTER
2551 010206 042702 020100 BIC #20100,R2 ;MASK OPI AND VPE
2552 010212 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2553 010214 001402 BEQ LT30X ;IF NOT: BR
2554 010216 004737 015206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2555 010222 004737 016576 LT30X: JSR PC,ITER ;GO SEE IF ITERATION
2556 010226 004737 015052 JSR PC,EORPA ;GO CLEAR GO BIT
2557 010232 004737 015666 JSR PC,DRVCLR
2558 010236 000137 002526 JMP TSCD2 ;RETURN TO SCHED
2559

```



```

2561
2562
2563
2564 010242 012737 025630 000616 LT31: MOV #MSLT31,EMADDR ;SET TEST HEADER
2565 010250 012737 010250 000706 LT31IT: MOV #LT31IT,SCOLP ;SET SCOPE ADDRESS
2566 010256 012737 027656 000666 MOV #TMS33A,ERADD ;SET ERROR MSG HDR
2567 010264 012737 000002 000602 MOV #2,ITAMT ;SET REDUCED ITER COUNT
2568 010272 004737 016752 JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2569 010276 005000 CLR R0
2570 010300 005300 1$: DEC R0
2571 010302 001376 BNE 1$ ;AWAIT OPI RESET
2572 010304 052777 000300 170230 BIS #300,@TC ;SET FORMAT
2573 010312 012777 000013 170214 MOV #13,@MR ;SET WAM 2
2574 010320 005077 170172 CLR @FC ;SET FRAME COUNT
2575 010324 012705 020000 MOV #20000,R5 ;SET TEST BIT (OPI)
2576 010330 012702 010346 MOV #2$,R2 ;SET RETURN ADDRESS FROM TIMER
2577 010334 004737 010546 JSR PC,TIMON ;START TIMER
2578 010340 012777 000061 170142 MOV #61,@C1 ;LOAD WRITE+GO
2579 010346 030577 170152 2$: BIT R5,@ER ;BRANCH WHEN OPI SETS
2580 010352 001002 BNE 3$
2581 010354 000163 010640 JMP TIMER(R3) ;GO TO TIMER & RETURN TO 2$ ABOVE
2582 010360 017702 170140 3$: MOV @ER,R2 ;GET ERROR REGISTER
2583 010364 020502 CMP R5,R2 ;SEE IF UNEXPECTED ERRORS
2584 010366 001403 BEQ 4$ ;IF NOT: BR
2585 010370 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
2586 010374 000453 BR LT31X
2587 010376 004737 010732 4$: JSR PC,TIMOK ;GO CHECK TIME FOR OPI TO SET
2588 010402 102450 BVS LT31X ;BRANCH IF TIME WAS INCORRECT
2589
2590 010404 012737 010420 000706 MOV #LT31A,SCOLP ;SET SCOPE LOOP
2591 010412 012737 027672 000666 MOV #TMS33B,ERADD ;SET ERROR MSG HEADER
2592 010420 004737 016752 LT31A: JSR PC,INIT3 ;GO INIT
2593 010424 005000 CLR R0
2594 010426 005300 1$: DEC R0 ;WAIT FOR OPI TO CLEAR
2595 010430 001376 BNE 1$
2596 010432 052777 000300 170102 BIS #300,@TC ;SET FORMAT
2597 010440 012777 000015 170066 MOV #15,@MR ;SET WRAP 3
2598 010446 012702 010470 MOV #2$,R2 ;SET RETURN ADDRESS FROM TIMER
2599 010452 012705 020000 MOV #20000,R5 ;SET TEST WORD
2600 010456 004737 010546 JSR PC,TIMON ;START TIMER
2601 010462 012777 000071 170020 MOV #71,@C1 ;LOAD READ+GO
2602 010470 030577 170030 2$: BIT R5,@ER ;BRANCH WHEN OPI SETS
2603 010474 001002 BNE 3$
2604 010476 000163 010640 JMP TIMER(R3) ;GO TO TIMER
2605 010502 017702 170016 3$: MOV @ER,R2 ;GET ERROR REGISTER
2606 010506 020502 CMP R5,R2 ;SEE IF UNEXPECTED ERRORS
2607 010510 001403 BEQ 4$
2608 010512 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
2609 010516 000402 BR LT31X ;EXIT TEST
2610 010520 004737 010732 4$: JSR PC,TIMOK ;GO CHECK TIME
2611 010524 004737 016576 LT31X: JSR PC,ITER ;GO SEE IF ITERATIONS
2612 010530 004737 015666 JSR PC,DRVCLR
2613 010534 012737 000020 000602 MOV #20,ITAMT
2614 010542 000137 002526 JMP TSCD2 ;RETURN TO SCHED
2615
2616
;ROUTINE TO START THE TIMER. THE TIMER IS AN OSCILLATOR IN THE MAINT-

```

```

2617                                     ;ENANCE REGISTER (BIT 6) THAT TOGGLES EVERY 56 (10) MICROSECONDS. THIS
2618                                     ;ROUTINE WAITS FOR THE OSCILLATOR TO TOGGLE AND RETURN WITH R3 INDICATING
2619                                     ;THE STATE OF THE OSCILLATOR.
2620 010546 005000 TIMON: CLR R0 ;CLEAR TICK COUNT
2621 010550 005001 CLR R1
2622 010552 012703 000024 MOV #24,R3 ;PRESET INDEX TO TIMER
2623 010556 032777 000100 167750 BIT #100,AMR ;BRANCH IF OSC CLEAR
2624 010564 001405 BEQ 2$
2625 010566 032777 000100 167740 1$: BIT #100,AMR ;WAIT FOR OSC TO CLEAR
2626 010574 001374 BNE 1$
2627 010576 000405 BR 4$ ;EXIT
2628
2629 010600 005403 2$: NEG R3 ;SET INDEX TO TIMER
2630 010602 032777 000100 167724 3$: BIT #100,AMR ;WAIT FOR OSC TO SET
2631 010610 001774 BEQ 3$
2632 010612 000207 4$: RTS PC ;RETURN
2633
2634                                     ;THIS ROUTINE TIMES AN EVENT. EACH TIME THE OSCILLATOR BIT CHANGES
2635                                     ;STATE THE TICK COUNT IN R1 & R0 IS INCREMENTED. THE ROUTINE IS CALLED
2636                                     ;USING R3 AS AN INDEX TO INDICATE THE OSCILLATORS PAST STATE. WHEN
2637                                     ;THE OSC BIT CHANGES STATE R3 IS NEGATED.
2638 010614 032777 000100 167712 TIMER1: BIT #100,AMR ;BRANCH IF OSC HAS CHANGED STATE
2639 010622 001406 BEQ TIMER
2640 010624 000112 JMP (R2) ;RETURN
2641 010640 010640 .=TIMER1+24
2642 010640 005403 TIMER: NEG R3 ;SET INDEX TO OTHER STATE
2643 010642 062700 000001 ADD #1,R0 ;INCREMENT TICK COUNT
2644 010646 005501 ADC R1
2645 010650 022701 000003 CMP #3,R1 ;BRANCH IF TIMER OVERFLOWS
2646 010654 001410 BEQ TIMOVF
2647 010656 000112 JMP (R2) ;RETURN
2648 010664 010664 .=TIMER+24
2649 010664 032777 000100 167642 TIMER0: BIT #100,AMR ;BRANCH IF OSC SET
2650 010672 001362 BNE TIMER
2651 010674 000112 JMP (R2) ;RETURN
2652
2653 010676 013704 000616 TIMOVF: MOV EMADDR,R4 ;TYPE TEST HEADER
2654 010702 004737 017724 JSR PC,TTOUT
2655 010706 013704 000666 MOV ERADD,R4 ;GET ERROR MSG ADDRESS
2656 010712 004737 017724 JSR PC,TTOUT ;AND TYPE IT
2657 010716 012704 027752 MOV #TMS33E,R4 ;TYPE
2658 010722 004737 017724 JSR PC,TTOUT ;'TIMER OVERFLOWED'
2659 010726 000137 010524 JMP LT31X ;GO EXIT TEST
2660
2661                                     ;ROUTINE TO CHECK IF TIME IS WITHIN LIMITS. IF NOT THE ROUTINE RETURNS
2662                                     ;WITH THE 'V' BIT SET. THE LIMITS WERE SLECTED BY DIVIDING THE TIME
2663                                     ;IN MICROSECONDS BY 448. THE LOWER LIMIT IS 5,500,000 USECS (5.5 SECS);
2664                                     ;THE UPPER LIMIT IS 9,500,000 USECS (9.5 SECS). THE 448 IS DERIVED FROM
2665                                     ;56 USECS/TICK TIMES THE DIVISION BY 8 BY THE TIMOK ROUTINE.
2666 010732 000240 TIMOK: NOP
2667 010734 006201 ASR R1 ;DIVIDE COUNT BY 8
2668 010736 006000 ROR R0
2669 010740 006201 ASR R1
2670 010742 006000 ROR R0
2671 010744 006201 ASR R1
2672 010746 006000 ROR R0

```


2673	010750	013701	001010		MOV	SLVTYP,R1		;++B GET SLAVE TYPE (0/1 = TE16/TU77)
2674	010754	006301			ASL	R1		;++B FORM INDEX
2675	010756	020061	011062		CMP	R0,200\$(R1)		;++B BRANCH IF GREATER THAN LOWER LIMIT(5.5 SECS)
2676	010762	101016			BHI	1\$		
2677	010764	013704	000616		MOV	EMADDR,R4		:GET ERROR MSG HEADER
2678	010770	004737	017724		JSR	PC,TTOUT		:TYPE ERROR MSG HEADER
2679	010774	013704	000666		MOV	ERADD,R4		:GET ERROR DESCRIPTOR MSG
2680	011000	004737	017724		JSR	PC,TTOUT		
2681	011004	012704	027705		MOV	#TMS33C,R4		:TYPE 'OCCURED TOO SOON'
2682	011010	004737	017724		JSR	PC,TTOUT		
2683	011014	000262			SEV			:SET 'V' TO INDICATE ERROR
2684	011016	000420			BR	2\$		
2685								
2686	011020	020061	011066	1\$:	CMP	R0,201\$(R1)		;++B BRANCH IF LESS THAN UPPER LIMIT(9.5 SECS)
2687	011024	003415			BLE	2\$		
2688	011026	013704	000616		MOV	EMADDR,R4		:GET ERROR MSG HEADER
2689	011032	004737	017724		JSR	PC,TTOUT		
2690	011036	013704	000666		MOV	ERADD,R4		
2691	011042	004737	017724		JSR	PC,TTOUT		:TYPE ERROR MSG HEADER
2692	011046	012704	027727		MOV	#TMS33D,R4		:TYPE 'OCCURED TOO LATE'
2693	011052	004737	017724		JSR	PC,TTOUT		
2694	011056	000262			SEV			
2695	011060	000207		2\$:	RTS	PC		
2696								
2697								
2698								
2699								
2700	011062	027764						
2701	011064	020622						
2702								
2703								
2704	011066	051325						
2705	011070	034774						

								;++B TABLE OF MIN AND MAX TIMES FOR OPI FOR TE16 AND TU77 SLAVES
								;++B MIN TIMES (5.5 SECS)
200\$:	.WORD	12276.						;++B TE16
	.WORD	8594.						;++B TU77
								;++B MAX TIMES (9.5 SECS)
201\$:	.WORD	21205.						;++B TE16
	.WORD	14844.						;++B TU77

2707
 2708
 2709
 2710
 2711
 2712
 2713
 2714
 2715
 2716
 2717
 2718
 2719
 2720
 2721
 2722
 2723
 2724
 2725
 2726
 2727
 2728
 2729
 2730
 2731
 2732
 2733
 2734
 2735
 2736
 2737
 2738
 2739

011072 012737 025664 000616
 011100 012737 011106 000706
 011106 004737 016752
 011112 013700 000660
 011116 012701 040000
 011122 032777 000004 167406
 011130 001402
 011132 052701 004000
 011136 005100
 011140 042700 177770
 011144 052700 000300
 011150 010077 167366
 011154 032777 002000 167354
 011162 001030
 011164 012777 000071 167316
 011172 017702 167326
 011176 030102
 011200 001011
 011202 012737 027772 000666
 011210 012737 000001 000712
 011216 004737 015220
 011222 000404
 011224 020102
 011226 001402
 011230 004737 015206
 011234 004737 016576
 011240 004737 015666
 011244 000137 002526

```

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


2741
 2742
 2743
 2744
 2745
 2746
 2747
 2748
 2749
 2750
 2751
 2752
 2753
 2754
 2755
 2756
 2757
 2758
 2759
 2760
 2761

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

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

011250	012737	025720	000616	LT33:	MOV	#MSLT33,EMADDR	:SET TEST HEADER
011256	012737	011264	000706		MOV	#LT33IT,SCOLP	:SET SCOPE ADDRESS
011264	004737	016752		LT33IT:	JSR	PC,INIT3	:INIT, SELECT DRIVE+SLAVE
011270	012777	000013	167236		MOV	#13,@MR	:SET WAM 2
011276	012777	177777	167212		MOV	#-1,@FC	:SET FCS
011304	012777	000031	167176		MOV	#31,@C1	:LOAD SPACE FORWARD+GO
011312	032777	020000	167202		BIT	#20000,@DS	:SEE IF PIP=1
011320	001010				BNE	LT33X	:IF SO: BR
011322	012737	027465	000666		MOV	#TMS14,ERADD	:SET ERROR CODE
011330	012737	000001	000712		MOV	#1,EXFL	:SET ERROR CODE
011336	004737	015220			JSR	PC,LTGERO	:GO PRINT ERROR
011342	004737	016576		LT33X:	JSR	PC,ITER	:GO SEE IF ITERATIONS
011346	000137	002526			JMP	TSCD2	:RETURN TO SCHED

```

2763
2764
2765
2766 011352 012737 027405 000666 LT34: MOV #TMS6,ERADD ;SET ERROR CODE
2767 011360 012737 025754 000616 MOV #MSLT34,EMADDR ;SET TEST HEADER
2768 011366 012700 000004 LT34IT: MOV #4,R0
2769 011372 004737 016752 LT34A1: JSR PC,INIT3 ;GO INIT, SELECT DRIVE+SLAVE
2770 011376 042777 003400 167136 BIC #3400,@TC ;SELECT NRZI
2771 011404 052777 001400 167130 BIS #1400,@TC
2772 011412 032777 000040 167102 LT34A: BIT #40,@DS ;SEE IF PES=0
2773 011420 001410 BEQ LT34B ;IF SO: BR
2774 011422 012737 000002 000712 MOV #2,EXFL ;SET RCVD-NOT EXPT
2775 011430 012737 011372 000706 MOV #LT34A1,SCOLP ;SET SCOPE ADDRESS
2776 011436 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2777 011442 004737 016766 LT34B: JSR PC,INIT4
2778 011446 032777 000040 167046 LT34C: BIT #40,@DS ;SEE IF PES=1
2779 011454 001010 BNE LT34X ;IF SO: BR
2780 011456 012737 011446 000706 MOV #LT34C,SCOLP ;SET SCOPE ADDRESS
2781 011464 012737 000001 000712 MOV #1,EXFL ;SET EXPT-NOT RCVD FLAG
2782 011472 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2783 011476 004737 016576 LT34X: JSR PC,ITER ;GO SEE IF ITERATION
2784 011502 000137 002526 LT34XX: JMP TSCD2 ;RETURN TO SCHED
  
```



```
2786  
2787  
2788  
2789 011506 012737 030015 000666 LT35: MOV #TMS37,ERADD  
2790 011514 012737 026010 000616 MOV #MSLT35,EMADDR  
2791 011522 004737 016752 LT35IT: JSR PC,INIT3 ;INIT SELECT DRIVE, SLAVE  
2792 011526 032777 000020 166766 1$: BIT #20,ADS ;SEE IF SDWN IS RESET  
2793 011534 001374 BNE 1$ ;IF NOT: BR  
2794 011536 052777 000300 166776 BIS #300,@TC ;SET FORMAT  
2795 011544 012777 000015 166762 MOV #15,@MR ;SET WAM 3  
2796 011552 012777 000071 166730 MOV #71,@C1 ;LOAD READ+GO  
2797 011560 032777 020000 166754 BIT #20000,@TC ;SEE IF SAC=0  
2798 011566 001410 BEQ LT35A ;IF SO: BR  
2799 011570 012737 000002 000712 MOV #2,EXFL ;SET RCV-NOT EXPT FLAG  
2800 011576 012737 011522 000706 MOV #LT35IT,SCOLP ;SET SCOPE ADDRESS  
2801 011604 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR  
2802 011610 004737 016752 LT35A: JSR PC,INIT3 ;INIT  
2803 011614 005277 166722 INC @TC ;BUMP SLAVE ADDRESS  
2804 011620 032777 020000 166714 BIT #20000,@TC ;SEE IF SAC=1  
2805 011626 001010 BNE LT35X ;IF SO: BR  
2806 011630 012737 011610 000706 MOV #LT35A,SCOLP ;SET SCOPE ADDRESS  
2807 011636 012737 000001 000712 MOV #1,EXFL ;SE EXPT-NOT RCVD FLAG  
2808 011644 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR  
2809 011650 004737 016576 LT35X: JSR PC,ITER  
2810 011654 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```

2812
2813
2814 ;LOGIC TEST 36: FRAME COUNTER STATUS(FCS)*****
2815 011660 012737 026055 000616 LT36: MOV #MSLT36,EMADDR
2816 011666 012737 030023 000666 MOV #TMS38,ERADD ;SET ERROR CODE
2817 011674 004737 016752 LT36IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2818 011700 032777 040000 166634 BIT #40000,@TC ;SEE IF FCS=0
2819 011706 001410 BEQ 1$ ;IF SO: BR
2820 011710 012737 011674 000706 MOV #LT36IT,SCOLP ;SET SCOPE ADDRESS
2821 011716 012737 000002 000712 MOV #2,EXFL ;SET RCVD-NOT EXPT
2822 011724 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2823 011730 004737 016752 1$: JSR PC,INIT3 ;INIT
2824 011734 005077 166556 CLR @FC ;WRITE TO FC
2825 011740 032777 040000 166574 BIT #40000,@TC ;SEE IF FCS=1
2826 011746 001010 BNE LT36X ;IF SO: BR
2827 011750 012737 011730 000706 MOV #1$,SCOLP ;SET SCOPE ADDRESS
2828 011756 012737 000001 000712 MOV #1,EXFL ;SET EXPT-NOT RCVD
2829 011764 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2830 011770 004737 016576 LT36X: JSR PC,ITER
2831 011774 000137 002526 JMP TSCD2 ;RETURN TO SCHED
  
```



```
2833
2834
2835
2836 012000 012737 026122 000616 LT37: MOV #MSLT37,EMADDR
2837 012006 012737 030031 000666 MOV #TMS39,ERADD ;SET ERROR CODE
2838 012014 004737 016752 LT37IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2839 012020 052777 000300 166514 BIS #300,@TC ;SET FORMAT
2840 012026 005777 166510 TST @TC ;SEE IF ACCL=1
2841 012032 100410 BMI LT37A ;IF SO: BR
2842 012034 012737 000001 000712 MOV #1,EXFL
2843 012042 012737 012014 000706 MOV #LT37IT,SCOLP ;SET SCOPE ADDRESS
2844 012050 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2845 012054 004737 016752 LT37A: JSR PC,INIT3 ;INIT
2846 012060 052777 000300 166454 BIS #300,@TC ;SET FORMAT
2847 012066 012777 000015 166440 MOV #15,@MR ;SET WAM 3
2848 012074 012777 000071 166406 MOV #71,@C1 ;LOAD READ+GO
2849 012102 012700 100000 MOV #100000,RO ;SET ACCL DELAY
2850 012106 005777 166430 LT37B: TST @TC ;SEE IF ACCL=0
2851 012112 100012 BPL LT37X ;IF SO: BR
2852 012114 005300 DEC RO
2853 012116 001373 BNE LT37B ;DELAY
2854 012120 012737 012054 000706 MOV #LT37A,SCOLP ;SET SCOPE ADDRESS
2855 012126 012737 000002 000712 MOV #2,EXFL
2856 012134 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2857 012140 004737 016576 LT37X: JSR PC,ITER
2858 012144 000137 002526 JMP TSCD2 ;RETURN TO SCHED
```

```

2860
2861
2862
2863 012150 012737 012164 000706 LT40: MOV #LT40IT,SCOLP ;SET SCOPE ADDRESS
2864 012156 012737 026170 000616 MOV #MSLT40,EMADDR
2865 012164 004737 016766 LT40IT: JSR PC,INIT4 ;INIT, SELECT DRIVE+SLAVE
2866 012170 005000 CLR RO
2867 012172 005300 1$: DEC RO
2868 012174 001376 BNE 1$ ;DELAY FOR OPI RESET
2869 012176 052777 002300 166336 BIS #2300,@TC
2870 012204 012777 000007 166322 MOV #7,@MR ;SET WAM 0
2871 012212 012777 000027 166270 MOV #27,@C1 ;LOAD WRITE TAPE MARK+GO
2872 012220 012700 100000 MOV #100000,RO ;SET DELAY
2873 012224 032777 000004 166270 2$: BIT #4,@DS ;SEE IF TM=1
2874 012232 001012 BNE LT40X ;IF SO: BR
2875 012234 005300 DEC RO
2876 012236 001372 BNE 2$ ;DELAY
2877 012240 012737 027363 000666 MOV #TMS3,ERADD
2878 012246 012737 000001 000712 MOV #1,EXFL
2879 012254 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2880 012260 004737 016576 LT40X: JSR PC,ITER
2881 012264 000137 002526 LT40XX: JMP TSCD2 ;RETURN TO SCHED
  
```



```

2883
2884
2885
2886 012270 012737 012304 000706 LT41: MOV #LT41IT,SCOLP ;SET SCOPE ADDRESS
2887 012276 012737 026235 000616 MOV #MSLT41,EMADDR
2888 012304 004737 016752 LT41IT: JSR PC,INIT3 ;INIT, SELECT DRIVE,SLAVE
2889 012310 052777 001700 166224 BIS #1700,@TC ;SET NRZ+NORMAL FORMAT
2890 012316 012777 177760 166172 MOV #-20,@FC ;SET FCS
2891 012324 012777 000007 166202 MOV #7,@MR ;SET WAM 0
2892 012332 012777 000027 166150 MOV #27,@C1 ;LOAD WRITE TAPE MARK+GO
2893 012340 005000
2894 012342 032777 000004 166152 1$: BIT #4,@DS ;SEE IF TM=1
2895 012350 001012 BNE 2$ ;IF SO: BR
2896 012352 005300 DEC R0
2897 012354 001372 BNE 1$ ;DELAY
2898 012356 012737 027363 000666 MOV #TMS3,ERADD ;SET ERROR CODE
2899 012364 012737 000001 000712 MOV #1,EXFL
2900 012372 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2901 012376 032777 002000 166120 2$: BIT #2000,@ER ;SEE IF ITM=1
2902 012404 001010 BNE 3$ ;IF SO: BR
2903 012406 012737 027626 000666 MOV #TMS30,ERADD ;SET ERROR CODE
2904 012414 012737 000001 000712 MOV #1,EXFL
2905 012422 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2906 012426 032777 000100 166070 3$: BIT #100,@ER ;SEE IF VPE=1
2907 012434 001011 BNE 4$ ;IF SO: BR
2908 012436 012737 027613 000666 MOV #TMS28,ERADD ;SET ERROR CODE
2909 012444 012737 000001 000712 MOV #1,EXFL
2910 012452 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
2911 012456 000410 BR LT41X
2912 012460 012701 002100 4$: MOV #2100,R1 ;SET EXPT ERROR BITS
2913 012464 017702 166034 MOV @ER,R2 ;GET ERROR REGISTER
2914 012470 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2915 012472 001402 BEQ LT41X ;IF NOT: BR
2916 012474 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
2917 012500 005002 LT41X: CLR R2 ;SET TIMER
2918 012502 032777 000200 166012 1$: BIT #200,@DS ;SEE IF DRY SET
2919 012510 001002 BNE 2$ ;IF SO: BR
2920 012512 005302 DEC R2 ;AWAIT DRY
2921 012514 001372 BNE 1$ ;DELAY
2922 012516 004737 016576 2$: JSR PC,ITER ;GO SEE IF ITERATIONS
2923 012522 004737 015666 JSR PC,DRVCLR ;GO DO DRIVE CLEAR
2924 012526 000137 002526 JMP TSCD2 ;RETURN TO SCHED
  
```

```

2926
2927
2928
2929
2930
2931
2932
2933 012532 012737 001700 000772 LT42: MOV #1700, UDES ;SET UNIT DESCRIPTION = NRZ
2934 012540 004737 015022 JSR PC, STATIC ;GO SEE IF STATIC ONLY
2935 012544 012700 001000 MOV #1000, R0
2936 012550 005300 1$: DEC R0
2937 012552 001376 BNE 1$ ;PAUSE
2938 012554 012737 026304 000616 MOV #MSLT42, EMADDR
2939 012562 012737 012570 000706 MOV #LT42IT, SCOLP ;SET SCOPE ADDRESS
2940 012570 004737 017002 LT42IT: JSR PC, INIT ;INIT SELECT DRIVE+SLAVE
2941 012574 012777 177770 165710 MOV #-10, @WC
2942 012602 012777 177760 165706 MOV #-20, @FC ;SET FC=20
2943 012610 012777 030040 165676 MOV #WDATA, @BA ;SET BUS ADDRESS
2944 012616 012777 000007 165710 MOV #7, @MR ;SET MM CODE
2945 012624 012777 000061 165656 MOV #61, @C1 ;LOAD WRITE+GO
2946 012632 005000 CLR R0
2947 012634 032777 000200 165660 LT42A: BIT #200, @DS ;SEE IF DRY=1
2948 012642 001002 BNE LT42B ;IF SO: BR
2949 012644 005300 DEC R0
2950 012646 001372 BNE LT42A ;DELAY
2951 012650 022777 000200 165646 LT42B: CMP #200, @ER ;SEE IF LRC ERROR ONLY
2952 012656 001007 BNE LT42B1 ;IF NOT: BR
2953 012660 017702 165644 MOV @CC, R2 ;GET CHECK CHAR
2954 012664 042702 177000 BIC #177000, R2 ;MASK CRC
2955 012670 022702 000777 CMP #777, R2 ;SEE IF SETUP CRC IS CORRECT
2956 012674 001410 BEQ LT42B2 ;IF SO: BR
2957 012676 004737 015206 LT42B1: JSR PC, LTGER3 ;ELSE PRINT ERROR SETUP
2958 012702 012704 023076 MOV #MSG55, R4
2959 012706 004737 017724 JSR PC, TTOUT ;PRINT SETUP ERROR MSG
2960 012712 000137 002526 JMP TSCD2 ;RETURN TO SCHED
2961 012716 004737 017002 LT42B2: JSR PC, INIT ;GO INIT
2962 012722 012777 177770 165562 MOV #-10, @WC ;SET WC
2963 012730 012777 177760 165560 MOV #-20, @FC ;SET FC
2964 012736 012777 030040 165550 MOV #WDATA, @BA ;SET BA
2965 012744 012777 000021 165562 MOV #21, @MR ;SET MM
2966 012752 012777 000061 165530 MOV #61, @C1 ;LOAD WRITE+GO
2967 012760 005000 CLR R0
2968 012762 032777 000200 165532 LT42C: BIT #200, @DS ;SEE IF DRY
2969 012770 001002 BNE LT42D ;IF SO: BR
2970 012772 005300 DEC R0
2971 012774 001372 BNE LT42C ;AWAIT DRY
2972 012776 005777 165522 LT42D: TST @ER ;SEE IF CRC=1
2973 013002 100411 BMI LT42E ;IF SO: BR
2974 013004 012737 030007 000666 MOV #TMS36, ERADD ;SET ERROR CODE
2975 013012 012737 000001 000712 MOV #1, EXFL
2976 013020 004737 015220 JSR PC, LTGER0 ;GO PRINT ERROR
2977 013024 000410 BR LT42X
2978 013026 012701 100200 LT42E: MOV #100200, R1 ;SET EXPT ERROR BITS
2979 013032 017702 165466 MOV @ER, R2 ;GET ERROR REGISTER
2980 013036 020102 CMP R1, R2 ;SEE IF UNEXPECTED ERRORS
2981 013040 001402 BEQ LT42X ;IF NOT: BR
  
```



```
2982 013042 004737 015206          JSR    PC,LTGER3      ;ELSE PRINT ERROR
2983 013046 004737 016576          JSR    PC,ITER        ;DO ITERATIONS
2984 013052 004737 015666          JSR    PC,DRVCLR      ;
2985 013056 000137 002526          JMP    TSCD2          ;RETURN TO SCHED
2986
2987                               ;LOGIC TEST 43: LONGITUALINAL REDUNDANCY(LRC)*****
2988
2989 013062 032777 000004 165446  LT43:  BIT    #4,@DT      ;++B BRANCH IF NOT A TE16
2990 013070 001114                BNE    LT43XX         ;++B
2991 013072 012737 001700 000772    MOV    #1700,UDES     ;SET UNIT DESCRIPTION = NRZ
2992 013100 004737 015022                JSR    PC,STATIC      ;GO SEE IF STATIC ONLY
2993 013104 012737 013120 000706    MOV    #LT43IT,SCOLP  ;SET SCOPE ADDRESS
2994 013112 012737 026340 000616    MOV    #MSLT43,EMADDR
2995 013120 004737 017002          LT43IT: JSR    PC,INIT   ;INIT, SELECT DRIVE+SLAVE
2996 013124 005001                CLR    R1
2997 013126 005301          1$:    DEC    R1        ;DELAY
2998 013130 001376                BNE    1$
2999 013132 012777 000023 165374    MOV    #23,@MR        ;SET MM
3000 013140 012777 177770 165344    MOV    #-10,@WC       ;SET WC
3001 013146 012777 177760 165342    MOV    #-20,@FC       ;SET FC
3002 013154 012777 030040 165332    MOV    #WDATA,@BA     ;SET BA
3003 013162 012777 000061 165320    MOV    #61,@C1        ;LOAD WRITE+GO
3004 013170 005000                CLR    R0
3005 013172 032777 000200 165322  LT43C: BIT    #200,@DS   ;SEE IF DRY
3006 013200 001002                BNE    LT43D         ;IF SO: BR
3007 013202 005300                DEC    R0
3008 013204 001372                BNE    LT43C         ;AWAIT DRY
3009 013206 032777 000200 165310  LT43D: BIT    #200,@ER   ;SEE IF LRC=1
3010 013214 001011                BNE    LT43E         ;IF SO: BR
3011 013216 012737 027577 000666    MOV    #TMS26,ERADD   ;SET ERROR CODE
3012 013224 012737 000001 000712    MOV    #1,EXFL
3013 013232 004737 015220                JSR    PC,LTGER0     ;GO PRINT
3014 013236 000425                BR     LT43X
3015 013240 017702 165270          LT43E: MOV    @MR,R2
3016 013244 042702 000177          BIC    #177,R2        ;MASK LRC
3017 013250 012701 157600          MOV    #157600,R1     ;SET EXPT LRC
3018 013254 020102          CMP    R1,R2         ;SEE IF EXPT = RCVD
3019 013256 001405          BEQ    LT43F         ;IF SO: BR
3020 013260 012737 023053 000666    MOV    #MSG53,ERADD   ;SET ERROR CODE
3021 013266 004737 016334                JSR    PC,LTGER1     ;PRINT ERROR
3022 013272 017702 165226          LT43F: MOV    @ER,R2   ;GET ERROR REGISTER
3023 013276 012701 000200          MOV    #200,R1       ;SET EXPT ERROR BITS
3024 013302 020102          CMP    R1,R2         ;SEE IF UNEXPECTED ERRORS
3025 013304 001402          BEQ    LT43X         ;IF NOT: BR
3026 013306 004737 015206          JSR    PC,LTGER3     ;ELSE PRINT ERROR
3027 013312 004737 016576          LT43X: JSR    PC,ITER
3028 013316 004737 015666          JSR    PC,DRVCLR
3029 013322 000137 002526          LT43XX: JMP    TSCD2 ;RETURN TO SCHED
```

```

;LOGIC TEST 44: PE CORRECTABLE DATA (CORR)*****
3031
3032
3033 013326 012737 002300 000772 LT44: MOV #2300,UDES ;SET UNIT DESCRIPTION = PE
3034 013334 004737 015022 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3035 013340 012737 026374 000616 MOV #MSLT44,EMADDR ;SET HEADER
3036 013346 012737 013354 000706 MOV #LT44IT,SCOLP ;SET SCOP
3037 013354 004737 017002 LT44IT: JSR PC,INIT ;GO INITIALIZE
3038 013360 012777 177600 165124 MOV #-200,@WC ;SET WC=200
3039 013366 012777 177400 165122 MOV #-400,@FC ;SET FC=400
3040 013374 012777 030040 165112 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
3041 013402 012777 000061 165100 MOV #61,@C1 ;LOAD WRITE AND GO
3042 013410 005000 CLR R0
3043 013412 005777 165100 LT44A: TST @FC ;SEE IF FC=0
3044 013416 001402 BEQ LT44A1 ;IF SO:BR
3045 013420 005300 DEC R0
3046 013422 001373 BNE LT44A ;AWAIT FC=0
3047 013424 012777 000021 165102 LT44A1: MOV #21,@MR ;SET MAINT MODE
3048 013432 005000 CLR R0
3049 013434 032777 000200 165060 LT44B: BIT #200,@DS ;SEE IF DRY
3050 013442 001002 BNE LT44C ;IF SO :BR
3051 013444 005300 DEC R0
3052 013446 001372 BNE LT44B ;AWAIT DRY
3053 013450 005777 165050 LT44C: TST @ER ;SEE IF CORR=1
3054 013454 100410 BMI LT44D ;IF SO: BR
3055 013456 012737 030000 000666 MOV #TMS35,ERADD ;ELSE SET ERROR CODE
3056 013464 012737 000001 000712 MOV #1,EXFL ;SET EXPT FLAG
3057 013472 004737 015220 JSR PC,LTGER0 ;GO PRINT ERROR
3058 013476 000240 LT44D: NOP
3059 013500 122777 000002 165022 LT44E: CMPB #2,@CC ;SEE IF DEAD TRACK BIT 1
3060 013506 001414 BEQ LT44F ;IF SO: BR
3061 013510 117702 165014 MOVB @CC,R2 ;ELSE SAVE RECVD
3062 013514 042702 177000 BIC #177000,R2 ;MASK OUT CRC
3063 013520 112701 000002 MOVB #2,R1 ;SAVE EXPT
3064 013524 012737 022462 000666 MOV #MSG42,ERADD ;SET ERROR CODE
3065 013532 004737 016334 JSR PC,LTGER1 ;GO PRINT ERROR
3066 013536 000410 BR LT44X
3067 013540 017702 164760 LT44F: MOV @ER,R2 ;GET ERROR REGISTER
3068 013544 012701 100000 MOV #100000,R1 ;SET EXPT ERROR BITS
3069 013550 020102 CMP R1,R2 ;SEE IF EXPT=RCVD
3070 013552 001402 BEQ LT44X ;IF SO: BR
3071 013554 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
3072 013560 004737 016576 LT44X: JSR PC,ITER ;GO SEE IF ITERATIONS
3073 013564 004737 015666 JSR PC,DRVCLR ;GO DO DRIVE CLEAR
3074 013570 000137 002526 LT44XX: JMP TSCD2 ;RETURN TO SCHED
    
```



```

3076
3077 ;LOGIC TEST 45: PE INCORRECTABLE DATA(INC)*****
3078
3079 013574 012737 002300 000772 LT45: MOV #2300,UDES ;SET UNIT DESCRIPTION = PE
3080 013602 004737 015022 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3081 013606 012737 026454 000616 MOV #MSLT45,EMADDR
3082 013614 012737 013622 000706 MOV #LT45IT,SCOLP
3083 013622 004737 017002 LT45IT: JSR PC,INIT ;INIT SELECT DRIVE SLAVE
3084 013626 012777 177600 164656 MOV #-200,@WC ;SET WC=200
3085 013634 012777 177400 164654 MOV #-400,@FC ;SET FC=400
3086 013642 012777 030040 164644 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
3087 013650 012777 000061 164632 MOV #61,@C1 ;LOAD WRITE+GO
3088 013656 005000 CLR R0
3089 013660 005777 164632 LT45E: TST @FC ;AWAIT FC=0
3090 013664 001402 BEQ LT45E1
3091 013666 005300 DEC R0
3092 013670 001373 BNE LT45E ;AWAIT FC=0
3093 013672 012777 000023 164634 LT45E1: MOV #23,@MR ;SET MAINT CODE
3094 013700 005000 CLR R0
3095 013702 032777 000200 164612 LT45A: BIT #200,@DS ;SEE IF DRY IS SET
3096 013710 001002 BNE LT45B ;IF SO: BR
3097 013712 005300 DEC R0
3098 013714 001372 BNE LT45A ;AWAIT DRY
3099 013716 032777 000100 164600 LT45B: BIT #100,@ER ;SEE IF INC=1
3100 013724 001010 BNE LT45D ;IF SO:BR
3101 013726 012737 027555 000666 MOV #TMS23,ERADD ;SET ERROR CODE
3102 013734 012737 000001 000712 MOV #1,EXFL
3103 013742 004737 015220 JSR PC,LTGERO ;GO PRINT ERROR
3104 013746 017702 164556 LT45D: MOV @CC,R2 ;GET CHECK CHAR
3105 013752 042702 177000 BIC #177000,R2 ;MASK CHECK CHAR
3106 013756 012701 000046 MOV #46,R1 ;SET EXPT CK
3107 013762 020102 CMP R1,R2 ;SEE IF EXPT = RCVD
3108 013764 001405 BEQ LT45F ;IF SO: BR
3109 013766 012737 023065 000666 MOV #MSG54,ERADD
3110 013774 004737 016334 JSR PC,LTGER1 ;ELSE GO PRINT ERROR
3111 014000 017702 164520 LT45F: MOV @ER,R2
3112 014004 042702 120600 BIC #120600,R2 ;MASK OPI ,NSG, CORR, AND PEF
3113 014010 012701 000100 MOV #100,R1 ;SET EXPT ERROR BITS
3114 014014 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3115 014016 001402 BEQ LT45X ;IF NOT: BR
3116 014020 004737 015206 JSR PC,LTGER3 ;ELSE PRINT ERROR
3117 014024 004737 016576 LT45X: JSR PC,ITER
3118 014030 004737 015666 JSR PC,DRVCLR
3119 014034 000137 002526 LT45XX: JMP TSCD2 ;RETURN TO SCHED
    
```

```

3121
3122                                     ;LOGIC TEST 46: PE FORMAT ERROR(PEF,NSG)*****
3123
3124 014040 012737 002300 000772 LT46:  MOV    #2300,UDES      ;SET UNIT DESCRIPTION = PE
3125 014046 004737 015022          JSR    PC,STATIC    ;GO SEE IF STATIC ONLY
3126 014052 012737 026536 000616          MOV    #MSLT46,EMADDR ;SET HEADER
3127 014060 012737 014066 000706          MOV    #LT46IT,SCOLP ;SET SCOPE ADDRESS
3128 014066 004737 017002          LT46IT: JSR   PC,INIT    ;INITIALIZE
3129 014072 012777 177770 164412          MOV    #-10,@WC     ;SET WC=10
3130 014100 012777 177760 164410          MOV    #-20,@FC     ;SET FC=20
3131 014106 012777 030040 164400          MOV    #WDATA,@BA   ;SET BA=START OF WRITE BUFFER
3132 014114 012777 000061 164366          MOV    #61,@C1     ;LOAD WRITE+GO
3133 014122 005777 164370          LT46A:  TST    @FC
3134 014126 001375          BNE    LT46A       ;AWAIT FC=0
3135 014130 032777 000100 164376 1$:    BIT    #100,@MR     ;WAIT FOR TAPE TO START WRITEING POSTAMBLE
3136 014136 001774          BEQ    1$         ;DELAY
3137 014140 032777 000100 164366 2$:    BIT    #100,@MR
3138 014146 001374          BNE    2$
3139 014150 012777 000027 164356          MOV    #27,@MR     ;SET MM CODE TO KILL PEF
3140 014156 005000          CLR    R0         ;INIT TIMING LOOP
3141 014160 032777 000200 164334 LT46B:  BIT    #200,@DS   ;SEE IF DRY SET
3142 014166 001002          BNE    LT46C     ;IF SO: BR
3143 014170 005300          DEC    R0
3144 014172 001372          BNE    LT46B     ;AWAIT DRY
3145 014174 032777 000200 164322 LT46C:  BIT    #200,@ER   ;SEE IF PEF SET
3146 014202 001011          BNE    LT46D     ;IF SO: BR
3147 014204 012737 027571 000666          MOV    #TMS25,ERADD ;SET ERROR TAG
3148 014212 012737 000001 000712          MOV    #1,EXFL    ;SET EXPT FLAG
3149 014220 004737 015220          JSR    PC,LTGER0  ;GO PRINT ERROR
3150 014224 000420          BR    LT46X
3151 014226 017702 164272          LT46D:  MOV    @ER,R2  ;GET ERROR REGISTER
3152 014232 042702 120100          BIC    #120100,R2 ;++B CLEAR CRC,OPI & INC BITS (MAY OR MAY NOT SET)
3153 014236 012701 000600          MOV    #600,R1   ;++B SET EXPT ERROR BITS (NSG + PEF)
3154 014242 032777 000004 164266          BIT    #4,@DT    ;++B BRANCH IF TE16
3155 014250 001402          BEQ    1$       ;++B
3156 014252 042701 000400          BIC    #400,R1   ;++B TU77 SHOULD NOT SET NSG BIT
3157 014256 020102 1$:    CMP    R1,R2    ;SEE IF UNEXPECTED ERRORS
3158 014260 001402          BEQ    LT46X     ;IF NOT: BR
3159 014262 004737 015206          JSR    PC,LTGER3  ;ELSE PRINT ERROR
3160 014266 004737 016576          LT46X:  JSR    PC,ITER
3161 014272 004737 015666          JSR    PC,DRVCLR
3162 014276 000137 002526          LT46XX: JMP    TSCD2 ;RETURN TO SCHED

```



```
3196
3197
3198
3199 014472 012737 026642 000616 LT50: MOV #MSLT50,EMADDR ;SET ERROR POINTER FOR STANDARD
3200 014500 012737 014506 000706 MOV #LT50IT,SCOLP
3201 014506 004737 016752 LT50IT: JSR PC,INIT3 ;SET SLAVE = NRZ
3202 014512 042777 003400 164022 BIC #3400,@TC ;CLEAR DENSITY BITS
3203 014520 052777 002300 164014 BIS #2300,@TC ;SET DENSITY = PE
3204 014526 012777 177770 163756 MOV #-10,@WC ;SET WORD COUNT
3205 014534 012777 177760 163754 MOV #-20,@FC ;SET FRAME COUNT
3206 014542 012777 030040 163744 MOV #WDATA,@BA ;SET BUS ADDRESS
3207 014550 012777 000013 163756 MOV #13,@MR ;SET WRAP 2
3208 014556 012777 000061 163724 MOV #61,@C1 ;LOAD WRITE COMMAND
3209 014564 000240 NOP
3210 014566 000240 NOP
3211 014570 000240 NOP
3212 014572 012701 004000 2$: MOV #4000,R1 ;SET EXPECTED RESULT
3213 014576 017702 163722 3$: MOV @ER,R2 ;GET ERROR REGISTER
3214 014602 030102 BIT R1,R2 ;BRANCH IF NEF BIT SET
3215 014604 001006 BNE 1$
3216 014606 012737 000001 000712 MOV #1,EXFL ;SET EXPECTED FLAG
3217 014614 004737 015220 JSR PC,LTGER0 ;PRINT ERROR
3218 014620 000404 BR LT50X
3219 014622 020102 1$: CMP R1,R2 ;BRANCH IF NO UNEXPECTED ERROR
3220 014624 001402 BEQ LT50X ;BITS WERE SET
3221 014626 004737 015206 JSR PC,LTGER3 ;PRINT ERROR MSG
3222 014632 004737 016576 LT50X: JSR PC,ITER ;+ITERATE TEST
3223 014636 004737 015666 JSR PC,DRVCLR ;RESET DRIVE
3224 014642 000137 002526 JMP TSCD2
3225
3226
```



```
3228
3229
3230 ;LOGIC TEST 51: NEF WHEN WRITING NRZ ON PE SELECTED SLAVE
3231 014646 012737 026722 000616 LT51: MOV #MSLT51,EMADDR ;+SET ERROR POINTER FOR STANDARD CONF.
3232 014654 012737 014662 000706 MOV #LT51IT,SCOLP ;SET SCOPE LOOP ADDRS.
3233 014662 004737 016766 LT51IT: JSR PC,INIT4 ;SET SLAVE = PE
3234 014666 042777 002300 163646 BIC #2300,@TC ;CLEAR DENSITY BITS
3235 014674 052777 001300 163640 BIS #1300,@TC ;SET DENSITY = NRZ
3236 014702 012777 177770 163602 MOV #-10,@WC ;SET WORD COUNT
3237 014710 012777 177760 163600 MOV #-20,@FC ;SET FRAME COUNT
3238 014716 012777 030040 163570 MOV #WDATA,@BA ;SET BUS ADDRESS
3239 014724 012777 000013 163602 MOV #13,@MR ;SET WRAP 2
3240 014732 012777 000061 163550 MOV #61,@C1 ;SET WRITE COMMAND AND GO
3241 014740 000240 NOP
3242 014742 000240 NOP
3243 014744 000240 NOP
3244 014746 012701 004000 MOV #4000,R1 ;SET EXPECTED RESULT
3245 014752 017702 163546 MOV @ER,R2 ;GET ERROR REGISTER
3246 014756 030102 BIT R1,R2 ;BRANCH IF NEF SET
3247 014760 001006 BNE 1$
3248 014762 012737 000001 000712 MOV #1,EXFL ;SET EXPECTED FLAG
3249 014770 004737 015220 JSR PC,LTGER0 ;PRINT ERROR MSG
3250 014774 000404 BR LT51X
3251 014776 020102 1$: CMP R1,R2 ;BRANCH IF NO UNEXPECTED
3252 015000 001402 BEQ LT51X ;ERROR BITS WERE SET
3253 015002 004737 015206 JSR PC,LTGER3
3254 015006 004737 016576 LT51X: JSR PC,ITER ;+ITERATE TEST
3255 015012 004737 015666 JSR PC,DRVCLR ;CLEAR DRIVE
3256 015016 000137 002526 JMP TSCD2 ;RETURN TO SCHEDULER
```

```
3258                                     ;STATIC TESTS ONLY SUBROUTINE*****
3259
3260 015022 005737 000734          STATIC: TST      STFLG      ;SEE IF SINGLE TEST ONLY
3261 015026 001006                BNE        1$        ;IF SO: BR
3262 015030 005737 001006                TST      STATC     ;SEE IF STATIC ONLY
3263 015034 001403                BEQ        1$        ;IF NOT: BR
3264 015036 005726                TST      (SP)+     ;RESET STACK
3265 015040 000137 002526          JMP        TSCD2    ;RETURN TO SCHEDULAR
3266 015044 005037 001002          1$: CLR      RDRVF
3267 015050 000207                RTS        PC      ;RETURN TO TEST
3268
```



```

3270
3271
3272
3273 015052 017700 163456      EORPA: MOV    @MR,R0      ;GET MAINT REG
3274 015056 042700 000036      BIC    #36,R0      ;CLEAR CURRENT OP CODE
3275 015062 052700 000024      BIS    #24,R0      ;SET EOR CLEAR OP CODE
3276 015066 010077 163442      MOV    R0,@MR      ;DO EOR
3277 015072 042777 000037 163434 BIC    #37,@MR      ;CLEAR EOR AND MM
3278 015100 005000
3279 015102 012701 000002      MOV    #2,R1
3280 015106 032777 000001 163374 EORP1: BIT    #1,@C1      ;SEE IF GO GONE
3281 015114 001430      BEQ    EORP2        ;IF SO: BR
3282 015116 005300
3283 015120 001372      DEC    R0
3284 015122 005301      BNE    EORP1        ;AWAIT GO RESET
3285 015124 001370
3286 015126 032777 020000 163434 BIT    #20000,@SWR   ;SEE IF ERROR PRINT INHIBIT
3287 015134 001020      BNE    EORP2        ;IF SO: BR
3288 015136 005737 000614      TST    HDRFL        ;SEE IF DONE HEADER
3289 015142 001004      BNE    EORP1A       ;IF SO: BR
3290 015144 013704 000616      MOV    EMADDR,R4
3291 015150 004737 017724      JSR    PC,TTOUT     ;PRINT HEADER
3292 015154 012704 030453      EORP1A: MOV   #WMSG31,R4
3293 015160 004737 017724      JSR    PC,TTOUT     ;PRINT EOR GO BIT ERROR
3294 015164 032777 100000 163376 BIT    #100000,@SWR ;SEE IF HALT ON ERROR
3295 015172 001401      BEQ    EORP2        ;IF NOT: BR
3296 015174 000000
3297 015176 000240      EORP2: NOP
3298 015200 005037 000672      EORPX: CLR    TEMP2 ;CLEAR FLAG
3299 015204 000207      RTS    PC           ;RETURN
3300
  
```

;LOGIC TEST ADDRESSING ERROR SUBROUTINE*****

3302									
3303									
3304	015206	005037	000712		LTGER3:	CLR	EXFL		
3305	015212	012737	023024	000666		MOV	#MSG51,ERADD		
3306	015220	012737	000001	000742	LTGER0:	MOV	#1,ADDFL		;SET NO ADDRESS FLAG
3307	015226	000240			LTGER:	NOP			
3308	015230	005037	000662			CLR	PFLG		;CLEAR PRINT FLAG
3309	015234	032777	020000	163326		BIT	#20000,@SWR		;SEE IF SHOULD PRINT
3310	015242	001112				BNE	LTGX		;IF NOT: BR
3311	015244	005737	000614		LTGA:	TST	HDRFL		;SEE IF PRINTED HEADER
3312	015250	001004				BNE	LTGA1		;IF SO: BR
3313	015252	013704	000616			MOV	EMADDR,R4		
3314	015256	004737	017724			JSR	PC,TTOUT		;PRINT TEST HEADER
3315	015262	012737	000001	000614	LTGA1:	MOV	#1,HDRFL		;SET HEADER FLAG
3316	015270	013704	000666			MOV	ERADD,R4		
3317	015274	004737	017724			JSR	PC,TTOUT		;PRINT CONDITION ERROR
3318	015300	005737	000742			TST	ADDFL		
3319	015304	001003				BNE	LTGA2		
3320	015306	010103				MOV	R1,R3		
3321	015310	004737	020052			JSR	PC,OCTP		;PRINT ADDRESS
3322	015314	005737	000712		LTGA2:	TST	EXFL		
3323	015320	001412				BEQ	LTGC		;IF NO STATUS: BR
3324	015322	012704	021315			MOV	#MSG6,R4		
3325	015326	022737	000001	000712		CMP	#1,EXFL		;EXPT-NOT RCVD
3326	015334	001402				BEQ	LTGB		
3327	015336	012704	021334			MOV	#MSG7,R4		;RCVD-NOT EXPT
3328	015342	004737	017724		LTGB:	JSR	PC,TTOUT		;PRINT STATUS
3329	015346	005237	000662		LTGC:	INC	PFLG		
3330	015352	005737	000742			TST	ADDFL		;SEE IF ADD TST
3331	015356	001430				BEQ	LTGD		;IF SO: BR
3332	015360	005737	000740			TST	T24FL		;SEE IF TEST 24
3333	015364	001423				BEQ	LTGC0		;IF NOT: BR
3334	015366	012704	030440			MOV	#WMSG27,R4		
3335	015372	004737	017724			JSR	PC,TTOUT		;PRINT DATA TAG
3336	015376	012704	021635			MOV	#MSG12,R4		
3337	015402	004737	017724			JSR	PC,TTOUT		;PRINT EXPT TAG
3338	015406	012703	177777			MOV	#-1,R3		
3339	015412	004737	020042			JSR	PC,OCTPE		;PRINT EXPT
3340	015416	012704	021644			MOV	#MSG13,R4		
3341	015422	004737	017724			JSR	PC,TTOUT		;PRINT RCVD TAG
3342	015426	010103				MOV	R1,R3		;GET RCVD
3343	015430	004737	020042			JSR	PC,OCTPE		;PRINT RCVD
3344	015434	004737	015532		LTGC0:	JSR	PC,REGP		;PRINT REGISTERS
3345	015440	032777	004000	163122	LTGD:	BIT	#4000,@SWR		
3346	015446	001010				BNE	LTGX		
3347	015450	012704	021706			MOV	#MSG16,R4		
3348	015454	004737	017724			JSR	PC,TTOUT		
3349	015460	013703	000676			MOV	ITCNT,R3		;PRINT ITERATION
3350	015464	004737	020052			JSR	PC,OCTP		
3351	015470	005777	163074		LTGX:	TST	@SWR		
3352	015474	100001				BPL	LTGXA		;IF NOT STOP ON ERROR: BR
3353	015476	000000				HALT			
3354	015500	005737	000662		LTGXA:	TST	PFLG		
3355	015504	001004				BNE	LTGX		;IF PRINTED: BR
3356	015506	032777	020000	163054		BIT	#20000,@SWR		
3357	015514	001653				BEQ	LTGA		

3358 015516 005037 000742
3359 015522 005037 000712
3360 015526 000137 016546
3361
3362
3363
3364 015532 000240
3365 015534 012704 022647
3366 015540 004737 017724
3367 015544 017703 162740
3368 015550 004737 020042
3369 015554 017703 162732
3370 015560 004737 020042
3371 015564 017703 162724
3372 015570 004737 020042
3373 015574 017703 162716
3374 015600 004737 020042
3375 015604 017703 162710
3376 015610 004737 020042
3377 015614 017703 162702
3378 015620 004737 020042
3379 015624 017703 162674
3380 015630 004737 020042
3381 015634 017703 162666
3382 015640 004737 020042
3383 015644 017703 162664
3384 015650 004737 020042
3385 015654 017703 162662
3386 015660 004737 020042
3387 015664 000207
3388
3389

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

;SUBROUTINE TO PRINT MAJOR REGISTERS*****

REGP: NOP
MOV #MSG46,R4
JSR PC,TTOUT ;PRINT REGISTER HEADER
MOV @C1,R3
JSR PC,OCTPE
MOV @WC,R3
JSR PC,OCTPE
MOV @BA,R3
JSR PC,OCTPE
MOV @FC,R3
JSR PC,OCTPE
MOV @CS,R3
JSR PC,OCTPE
MOV @DS,R3
JSR PC,OCTPE ;PRINT REGISTERS
MOV @ER,R3
JSR PC,OCTPE
MOV @AS,R3
JSR PC,OCTPE
MOV @MR,R3
JSR PC,OCTPE
MOV @TC,R3
JSR PC,OCTPE
RTS PC

```

3391                                     ;DRIVE CLEAR SUBROUTINE*****
3392
3393 015666 000240          DRVCLR: NOP
3394 015670 012704 040000      MOV      #40000,R4
3395 015674 005304          DCD:    DEC      R4
3396 015676 001376          BNE     DCD          ;DELAY
3397 015700 005037 000662      CLR     PFLG
3398 015704 004737 016122      JSR     PC,ATTN     ;GO SEE OF ATTN SET
3399 015710 012777 000011 162572  MOV     #11,@C1     ;ISSUE DRIVE CLEAR
3400 015716 005000          CLR     R0
3401 015720 032777 000200 162574  DCA:    BIT     #200,@DS     ;SEE IF DRY
3402 015726 001002          BNE     DCA0
3403 015730 005300          DEC     R0
3404 015732 001372          BNE     DCA          ;WAIT FOR DRY
3405 015734 032777 040000 162560  DCA0:   BIT     #40000,@DS     ;SEE IF ERR RESET
3406 015742 001022          BNE     DCE          ;IF NOT: BR
3407 015744 005777 162554      TST     @ER          ;SEE IF ERROR REGISTER RESET
3408 015750 001017          BNE     DCE          ;IF NOT: BR
3409 015752 005777 162544      TST     @DS          ;SEE IF ATA RESET
3410 015756 100414          BMI     DCE          ;IF NOT: BR
3411 015760 012703 000001      MOV     #1,R3        ;SET TEST BIT
3412 015764 013704 000620      MOV     DRVN,R4      ;GET DRIVE NUMBER & BRANCH
3413 015770 001403          BEQ     DCC
3414 015772 006303          DCB:    ASL     R3        ;POSITION TEST BIT PER DRIVE NUMBER
3415 015774 005304          DEC     R4
3416 015776 001375          BNE     DCB          ;SEE IF DONE
3417 016000 030377 162522      DCC:    BIT     R3,@AS     ;SEE IF ATTN IS RESET
3418 016004 001001          BNE     DCE          ;IF NOT: BR
3419 016006 000207          RTS     PC           ;RETURN
3420
3421 016010 000240          DCE:    NOP
3422 016012 032777 020000 162550  BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
3423 016020 001017          BNE     DCEX          ;IF SO: BR
3424 016022 005737 000614      TST     HDRFL        ;SEE IF PRINT HEADER
3425 016026 001004          BNE     DCEA          ;IF NOT: BR
3426 016030 013704 000616      MOV     EMADDR,R4
3427 016034 004737 017724      JSR     PC,TTOUT      ;PRINT HEADER
3428 016040 012704 022753      DCEA:   MOV     #MSG47,R4
3429 016044 004737 017724      JSR     PC,TTOUT      ;PRINT DRIVE CLEAR ERROR
3430 016050 004737 015532      JSR     PC,REGP       ;PRINT REGISTERS
3431 016054 005237 000662      INC     PFLG         ;SET PRINTED FLAG
3432 016060 005777 162504      DCEX:   TST     @SWR     ;SEE IF HALT ON ERROR
3433 016064 100001          BPL     DCEXA        ;IF NOT: BR
3434 016066 000000          HALT
3435 016070 005737 000662      DCEXA:  TST     PFLG     ;SEE IF HAVE PRINTED
3436 016074 001004          BNE     DCEXX        ;IF SO: BR
3437 016076 032777 020000 162464  BIT     #20000,@SWR     ;BRANCH IF ERROR
3438 016104 001741          BEQ     DCE          ;PRINTOUT DESIRED
3439 016106 000240          DCEXX:  NOP
3440 016110 012737 015666 000706  MOV     #DRVCLR,SCOLP ;SET SCOPE LOOP ADDRESS
3441 016116 000137 016546      JMP     SCOPE        ;GO DO SCOPE LOOP

```



```

3443                                     ;COMPOSITE ERROR CHECK SUBROUTINE*****
3444
3445 016122 000240          ATTN:  NOP
3446 016124 005777 162372      TST      @DS          ;SEE IF ATA SET
3447 016130 001004          BNE      ATTA          ;IF SO: BR
3448 016132 012737 022311 000674  MOV      #MSG32,TEMP3
3449 016140 000427          BR       ATTP          ;ELSE PRINT ERROR
3450 016142 032777 040000 162352  ATTA:  BIT      #40000,@DS  ;SEE IF COMPOSITE ERROR SET
3451 016150 001004          BNE      ATTB          ;IF SO: BR
3452 016152 012737 022273 000674  MOV      #MSG31,TEMP3
3453 016160 000417          BR       ATTP          ;ELSE PRINT ERROR
3454 016162 012703 000001      ATTB:  MOV      #1,R3    ;SET TEST BIT
3455 016166 012737 022327 000674  MOV      #MSG33,TEMP3
3456 016174 013704 000620      MOV      DRVN,R4      ;GET DRIVE NUMBER & BRANCH
3457 016200 001403          BEQ      ATTD          ;IF DRIVE 0
3458 016202 006303          ATTC:  ASL      R3      ;POSITION TEST BIT
3459 016204 005304          DEC      R4          ;SEE IF DONE
3460 016206 001375          BNE      ATTC          ;IF NOT: BR
3461 016210 030377 162312      ATTD:  BIT      R3,@AS  ;SEE IF ATTEN SUMMARY SET
3462 016214 001401          BEQ      ATTP          ;IF NOT: BR
3463 016216 000207          RTS      PC          ;ELSE RETURN
3464 016220 032777 020000 162342  ATTP:  BIT      #20000,@SWR ;SEE IF PRINT INHIBIT
3465 016226 001021          BNE      ATTX          ;IF SO: BR
3466 016230 005737 000614      TST      HDRFL        ;SEE IF DONE HEADER
3467 016234 001004          BNE      ATTPA         ;IF SO: BR
3468 016236 013704 000616      MOV      EMADDR,R4
3469 016242 004737 017724      JSR      PC,TTOUT      ;PRINT HEADER
3470 016246 013704 000674      ATTPA: MOV      TEMP3,R4
3471 016252 004737 017724      JSR      PC,TTOUT      ;PRINT ERROR TYPE
3472 016256 004737 015532      JSR      PC,REGP       ;PRINT REGISTERS
3473 016262 005237 000662      INC      PFLG          ;SET PRINT FLAG
3474 016266 005237 000614      INC      HDRFL        ;SET HEADER FLAG
3475 016272 005777 162272      ATTX:  TST      @SWR    ;SEE IF HALT ON ERROR
3476 016276 100001          BPL      ATTXA         ;IF NOT: BR
3477 016300 000000          HALT
3478 016302 005737 000662      ATTXA: TST      PFLG     ;SEE IF DONE PRINT
3479 016306 001004          BNE      ATTX          ;IF SO: BR
3480 016310 032777 020000 162252  BIT      #20000,@SWR  ;BRANCH IF NO ERROR
3481 016316 001740          BEQ      ATTP          ;PRINTOUT DESIRED
3482 016320 005037 000662      ATTXX: CLR      PFLG    ;CLEAR PRINT FLAG
3483 016324 000207          RTS      PC          ;RETURN
  
```

```

3485                                     ;LOGIC TEST REGISTER BIT ERROR SUBROUTINE*****
3486
3487 016326 012737 000001 000732 LTGER2: MOV #1,PEXFL ;SET FLAG
3488 016334 000240 LTGER1: NOP
3489 016336 005037 000662 CLR PFLG ;CLEAR PRINT FLAG
3490 016342 032777 020000 162220 BIT #20000,@SWR ;BRANCH IF ERROR
3491 016350 001055 BNE LTG1X ;PRINTOUT DESIRED
3492 016352 005737 000614 LTG1A: TST HDRFL ;SEE IF PRINT HEADER
3493 016356 001004 BNE LTG1B ;IF NOT: BR
3494 016360 013704 000616 MOV EMADDR,R4
3495 016364 004737 017724 JSR PC,TTOUT ;PRINT HEADER
3496 016370 012737 000001 000614 LTG1B: MOV #1,HDRFL ;SET FLAG
3497 016376 013704 000666 MOV ERADD,R4
3498 016402 004737 017724 JSR PC,TTOUT ;PRINT ERROR CODE
3499 016406 005737 000732 TST PEXFL ;SEE IF PRINT EXPT-RCVD
3500 016412 001016 BNE LTG1T ;IF NOT: BR
3501 016414 012704 021635 MOV #MSG12,R4
3502 016420 004737 017724 JSR PC,TTOUT ;PRINT EXPT TAG
3503 016424 010103 MOV R1,R3
3504 016426 004737 020052 JSR PC,OCTP ;PRINT EXPT
3505 016432 012704 021644 MOV #MSG13,R4
3506 016436 004737 017724 JSR PC,TTOUT ;PRINT RCVD TAG
3507 016442 010203 MOV R2,R3
3508 016444 004737 020052 JSR PC,OCTP ;PRINT RCVD
3509 016450 032777 004000 162112 LTG1T: BIT #4000,@SWR
3510 016456 001010 BNE LTG1C
3511 016460 012704 021706 MOV #MSG16,R4
3512 016464 004737 017724 JSR PC,TTOUT
3513 016470 013703 000676 MOV ITCNT,R3
3514 016474 004737 020052 JSR PC,OCTP ;PRINT ITERATION
3515 016500 005237 000662 LTG1C: INC PFLG
3516 016504 000240 LTG1X: NOP
3517 016506 005777 162056 TST @SWR
3518 016512 100001 BPL LTG1X1 ;IF NOT STOP ON ERROR: BR
3519 016514 000000 HALT
3520 016516 005737 000662 LTG1X1: TST PFLG
3521 016522 001004 BNE LTG1XX ;IF HAVE PRINTED: BR
3522 016524 032777 020000 162036 BIT #20000,@SWR
3523 016532 001707 BEQ LTG1A
3524 016534 000240 LTG1XX: NOP
3525 016536 005037 000732 CLR PEXFL ;CLEAR EXPT-RCVD FLAG
3526 016542 000137 016546 JMP SCOPE ;GO TO SCOPE
3527
3528
3529                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
3530
3531 016546 000240 SCOPE: NOP
3532 016550 032777 040000 162012 BIT #40000,@SWR ;SEE IF LOOP ON ERROR
3533 016556 001001 BNE 1$ ;IF SO: BR
3534 016560 000207 RTS PC ;ELSE EXIT
3535 016562 000240 1$: NOP
3536 016564 005726 TST (SP)+ ;RESET STACK
3537 016566 000240 NOP
3538 016570 000240 NOP
3539 016572 000177 162110 JMP @SCOLP ;LOOP ON ERROR
3540

```



```

3541                                     ;TEST ITERATION SUBROUTINE*****
3542
3543 016576 032777 004000 161764 ITER: BIT #4000,@SWR ;SEE IF ITERATIONS
3544 016604 001403 BEQ 2$ ;IF SO: BR
3545 016606 005037 000676 1$: CLR ITCNT ;CLEAR ITERATION COUNTER
3546 016612 000207 RTS PC ;ELSE EXIT
3547 016614 005737 001014 2$: TST PCNTR ;NO SUBTEST ITERATIONS ON FIRST PASS
3548 016620 001772 BEQ 1$
3549 016622 005237 000676 INC ITCNT ;BUMP COUNTER
3550 016626 023737 000676 000602 CMP ITCNT,ITAMT ;SEE IF DONE ALL
3551 016634 001764 BEQ 1$ ;IF SO: BR
3552 016636 005726 TST (SP)+ ;RESET STACK
3553 016640 017700 162044 MOV @ITRLP,R0 ;SET ITERATION POINTER
3554 016644 000110 JMP (R0) ;GO ITERATE
3555
3556                                     ;MANUAL INTERVENTION INHIBIT*****
3557
3558 016646 000240 INMT: NOP
3559 016650 012704 022477 MOV #MSG43,R4
3560 016654 004737 017724 JSR PC,TTOUT ;GO PRINT INHIB MSG
3561 016660 000000 HALT
3562 016662 000137 002526 JMP TSCD2 ;RETURN TO SCHED
3563
3564                                     ;NON-STANDARD MODE TEST HANDLER
3565
3566
3567 016666 010046 NOST: MOV R0,-(SP) ;+SAVE R0
3568 016670 012700 000240 MOV #240,R0 ;+SET UP INDEX
3569 016674 013760 001326 001056 MOV TADX,TSTTBL(R0) ;+
3570 016702 005720 TST (R0)+
3571 016704 012737 000047 001330 MOV #47,TLAST ;+SET END OF TEST
3572 016712 013760 001330 001056 MOV TLAST,TSTTBL(R0) ;+SET LAST TEST NUMBER
3573 016720 012600 MOV (SP)+,R0 ;+RESTORE R0
3574 016722 000207 RTS PC ;+RETURN
3575
    
```

```

3577
3578                                     ;INITIALIZE SUBROUTINE*****
3579
3580 016724 000240 INIT1: NOP
3581 016726 012777 000040 161564 MOV #40,@CS ;INIT
3582 016734 013777 000620 161556 INIT2: MOV DRVN,@CS ;SELECT DRIVE
3583 016742 013777 000660 161572 MOV SLVN,@TC ;SELECT SLAVE
3584 016750 000207 RTS PC ;RETURN
3585
3586 ;ROUTINES TO INITIALIZE SLAVE. THESE ROUTINES PLACE THE SLAVE
3587 ;IN PROPER STATUS FOR THE CALLING TEST. INIT3 PLACES THE SLAVE IN
3588 ;NRZ MODE AND OFF BOT; INIT4 PLACES THE SLAVE IN PE MODE AND OFF
3589 ;BOT. IF THE SLAVE IS IN THE PROPER STATUS ON ENTRY NO ACTION IS TAKEN.
3590
3591 ;SET SLAVE IN NRZ OFF BOT
3592 016752 013746 000772 INIT3: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
3593 016756 012737 001400 000772 MOV #1400,UDES ;SET UNIT DESCRIPTION = NRZ
3594 016764 000410 BR INIT5 ;GO TO INIT5 ROUTINE
3595
3596 ;SET SLAVE IN PE OFF BOT
3597 016766 013746 000772 INIT4: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
3598 016772 012737 002000 000772 MOV #2000,UDES ;SET UNIT DESCRIPTION = PE
3599 017000 000402 BR INIT5 ;GO DO IT
3600
3601 ;THIS ROUTINE IS ENTERED AT INIT WHEN THE CALLER HAS SETUP UDES.
3602 ;IT IS ENTERED AT INIT5 WHEN EITHER INIT3 OR INIT4 HAS SET UP UDES.
3603 017002 013746 000772 INIT: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
3604 017006 012777 000040 161504 INIT5: MOV #40,@CS ;INIT CONTROLLER
3605 017014 013777 000620 161476 MOV DRVN,@CS ;SELECT TM03 DRIVE
3606 017022 013777 000660 161512 MOV SLVN,@TC ;SELECT TE16 SLAVE
3607 017030 013746 000772 MOV UDES,-(SP) ;GET SLAVE DESCRIPTION
3608 017034 042716 174377 BIC #174377,(SP) ;CLEAR ALL BUT DENSITY SELECT BITS
3609 017040 022726 001400 CMP #1400,(SP)+ ;BRANCH IF REQUESTING PE MODE
3610 017044 001005 BNE 1$
3611 017046 032777 000040 161446 BIT #40,@DS ;BRANCH IF SLAVE IS IN NRZ MODE
3612 017054 001420 BEQ 4$ ;(PES = 0)
3613 017056 000404 BR 2$
3614 017060 032777 000040 161434 1$: BIT #40,@DS ;BRANCH IF SLAVE IS IN PE MODE
3615 017066 001013 BNE 4$
3616 017070 012777 000007 161412 2$: MOV #7,@C1 ;REWIND SLAVE
3617 017076 032777 000200 161416 20$: BIT #200,@DS ;WAIT FOR READY
3618 017104 001774 BEQ 20$
3619 017106 032777 020000 161406 3$: BIT #20000,@DS ;WAIT UNTIL PIP CLEARS
3620 017114 001374 BNE 3$
3621 017116 053777 000772 161416 4$: BIS UDES,@TC ;LOAD SLAVE DESCRIPTION
3622 017124 032777 000002 161370 BIT #2,@DS ;BRANCH IF NOT AT BOT
3623 017132 001407 BEQ 6$
3624 017134 012777 000025 161346 MOV #25,@C1 ;ERASE TO GET OFF BOT
3625 017142 032777 000200 161352 5$: BIT #200,@DS ;WAIT FOR READY
3626 017150 001774 BEQ 5$
3627 017152 032777 000020 161342 6$: BIT #20,@DS ;WAIT FOR SETTLEDOWN TO CLEAR
3628 017160 001374 BNE 6$
3629 017162 012777 000011 161320 MOV #11,@C1 ;RESET DRIVE
3630 017170 012637 000772 MOV (SP)+,UDES ;RESTORE UNIT DESCRIPTION
3631 017174 000207 RTS PC ;RETURN
3632
    
```



```

3633
3634
3635          ;MANUAL INSTRUCTION SUBROUTINE*****
3636
3637 017176 000240      INST:  NOP
3638 017200 004737 017724  JSR    PC,TTOUT      ;PRINT INSTRUCTION
3639 017204 012704 027002  MOV    #MMSG0,R4
3640 017210 004737 017724  JSR    PC,TTOUT      ;PRINT REPLY
3641 017214 012705 000674  MOV    #TEMP3,R5
3642 017220 012701 000001  MOV    #1,R1
3643 017224 012702 177777  MOV    #-1,R2
3644 017230 012703 000000  MOV    #0,R3
3645 017234 004737 017402  JSR    PC,TTR        ;AWAIT REPLY
3646 017240 000240      NOP
3647 017242 000207      RTS    PC            ;EXIT
3648
3649          ;MAG TAPE INTERRUPT HANDLER*****
3650
3651 017244 000240      MTINT: NOP
3652 017246 013716 000664  MOV    RTRN,(SP)    ;SET RETURN FROM INTERUPT ADDRESS
3653 017252 000002      RTI                ;RETURN
3654
3655          ;TTY INTERRUPT HANDLER*****
3656
3657 017254 017746 161314  TTINT: MOV    @TKB,-(SP)  ;GET CHARACTER
3658 017260 042716 000200  BIC    #200,(SP)    ;CLEAR PARITY BIT
3659 017264 122716 000003  CMPB   #3,(SP)      ;BRANCH IF NOT CONTROL C
3660 017270 001010      BNE    1$
3661 017272 005737 001420  TST    CHNFLG        ;INHIBIT ^C IF IN CHAIN MODE
3662 017276 001005      BNE    1$
3663 017300 005077 161262  CLR    @PSW          ;CLEAR PSW
3664 017304 000005      RESET
3665 017306 000137 000200  JMP    @#200          ;RESTART
3666 017312 122716 000001  1$:  CMPB   #1,(SP)      ;BRANCH IF NOT ^A
3667 017316 001017      BNE    2$
3668 017320 022737 000176 000570  CMP    #SWREG,SWR    ;BRANCH IF USING HARWARE SWR
3669 017326 001016      BNE    3$
3670 017330 012737 177570 000570  MOV    #177570,SWR   ;INVOKE HARDWARE SWR
3671 017336 004737 020576  JSR    PC,.SAVE      ;SAVE REGISTERS ON THE STACK
3672 017342 012704 023474  MOV    #MSG63,R4     ;TYPE 'HARDWARE SWR IN USE'
3673 017346 004737 017724  JSR    PC,TTOUT
3674 017352 004737 020620  JSR    PC,.RESTORE
3675 017356 122716 000007  2$:  CMPB   #7,(SP)      ;BRANCH IF NOT ^G
3676 017362 001005      BNE    4$
3677 017364 012737 000176 000570  3$:  MOV    #SWREG,SWR    ;INVOKE SOFTWARE SWR
3678 017372 004737 020500  JSR    PC,GTSWR      ;GET SWITCHES
3679 017376 005726  4$:  TST    (SP)+         ;POP CHARACTER OFF STACK
3680 017400 000002      RTI                ;RETURN
3681

```

```

3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700 017402 010146
3701 017404 011601
3702 017406 005037 000670
3703 017412 005000
3704 017414 004737 017662
3705 017420 122737 000003 000612
3706 017426 001003
3707 017430 000005
3708 017432 000137 000200
3709 017436 122737 000015 000612
3710 017444 001004
3711 017446 005737 000670
3712 017452 001471
3713 017454 000457
3714 017456 122737 000025 000612
3715 017464 001005
3716 017466 012704 023422
3717 017472 004737 017724
3718 017476 000742
3719 017500 122737 000177 000612
3720 017506 001012
3721 017510 000241
3722 017512 006000
3723 017514 006200
3724 017516 006200
3725 017520 012704 023424
3726 017524 004737 017724
3727 017530 005201
3728 017532 000730
3729 017534 122737 000060 000612
3730 017542 101402
3731 017544 000137 017642
3732 017550 122737 000070 000612
3733 017556 101002
3734 017560 000137 017642
3735 017564 005237 000670
3736 017570 006300
3737 017572 006300
3738 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$: MOV (SP), R1 ;RESTORE CHARACTER COUNT (FOR ^U)
CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
RO
1$: 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 RO ;REMOVE LAST CHAR
ASR RO
ASR RO
MOV #MSG60, R4 ;TYPE '\ '
JSR PC, TTOUT ;DECREMENT CHARS RECEIVED COUNT
INC R1
BR 1$
3$: CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0
BLOS 4$ ;IF NOT: BR
JMP TINER ;ELSE GO TO ERROR
4$: CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7
BHI 5$ ;IF NOT: BR
JMP TINER ;ELSE GO TO ERROR
5$: INC TEMP1 ;SET FIRST CHARACTER FLAG
ASL RO ;SHIFT 3 LEFT
ASL RO
ASL RO
  
```


3739	017576	042737	177770	000612	BIC	#177770,TIB	:STRIP ASCII
3740	017604	053700	000612		BIS	TIB,R0	:LOAD CHARACTER
3741	017610	005301			DEC	R1	:SEE IF DONE
3742	017612	001300			BNE	1\$:IF NOT: BR
3743	017614	020002		6\$:	CMP	R0,R2	:SEE IF EXCEEDED MAXIMUM LIMIT
3744	017616	101402			BLOS	7\$:IF NOT: BR
3745	017620	000137	017642		JMP	TINER	:ELSE GO TO ERROR
3746	017624	020300		7\$:	CMP	R3,R0	:SEE IF BELOW MINIMUM LIMIT
3747	017626	101402			BLOS	8\$:IF NOT: BR
3748	017630	000137	017642		JMP	TINER	:ELSE GO TO ERROR
3749	017634	010015		8\$:	MOV	R0,(R5)	:LOAD VALUE
3750	017636	005726		9\$:	TST	(SP)+	:POP CHAR COUNT OFF STACK
3751	017640	000207			RTS	PC	:EXIT
3752							

```

3754
3755 ;TTY ENTRY ERROR SUBROUTINE*****
3756
3757 017642 012704 022437 T1NER: MOV #MSG40,R4
3758 017646 004737 017724 JSR PC,TTOUT ;PRINT?
3759 017652 005726 TST (SP)+ ;POP CHAR COUNT OFF STACK
3760 017654 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
3761 017660 000207 RTS PC ;REDO VALUE ENTRY
3762
3763 ;TTY READ SUBROUTINE*****
3764
3765 017662 005277 160704 TTIN: INC @TKS
3766 017666 105777 160700 1$: TSTB @TKS
3767 017672 100375 BPL 1$
3768 017674 017737 160674 000612 MOV @TKB,TIB
3769 017702 042737 000200 000612 BIC #200,TIB ;STRIP PARITY BIT
3770 017710 013737 000612 000610 MOV TIB,TOB ;MOVE CHAR TO TTY OUTPUT BFR
3771 017716 004737 020024 JSR PC,TOG ;AND ECHO IT
3772 017722 000207 RTS PC
3773
3774 ;TTY OUTPUT SUBROUTINE*****
3775
3776 017724 112437 000610 TTOUT: MOVB (R4)+,TOB
3777 017730 122737 000043 000610 CMPB #43,TOB
3778 017736 001440 BEQ TEX
3779 017740 122737 000045 000610 CMPB #45,TOB
3780 017746 001403 BEQ 1$
3781 017750 004737 020024 JSR PC,TOG
3782 017754 000763 BR TTOUT
3783 017756 112737 000015 000610 1$: MOVB #15,TOB
3784 017764 004737 020024 JSR PC,TOG
3785 017770 012703 000004 MOV #4,R3
3786 017774 005037 000610 2$: CLR TOB
3787 020000 004737 020024 JSR PC,TOG
3788 020004 005303 DEC R3
3789 020006 001372 BNE 2$ ;DO FILLERS
3790 020010 112737 000012 000610 MOVB #12,TOB
3791 020016 004737 020024 JSR PC,TOG
3792 020022 000740 BR TTOUT
3793 020024 105777 160546 TOG: TSTB @TPS
3794 020030 100375 BPL TOG
3795 020032 113777 000610 160540 MOVB TOB,@TPB
3796 020040 000207 TEX: RTS PC
3797
3798
3799 ;OCTAL OUTPUT SUBROUTINE*****
3800
3801 020042 012737 000001 020272 OCTPE: MOV #1,OFL
3802 020050 000402 BR OCTPE1
3803 020052 005037 020272 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
3804 020056 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
3805 020060 001006 BNE OCTP0 ;IF NOT ZERO: BR
3806 020062 005737 020272 TST OFL ;SEE IF PRINT ALL 0
3807 020066 001003 BNE OCTP0 ;IF SO: BR
3808 020070 004737 020252 JSR PC,OCTPG1 ;ELSE PRINT ZERO
3809 020074 000447 BR OCTP3 ;SPACE AND EXIT

```


3810	020076	032704	100000		OCTP0:	BIT	#100000,R4	;SEE IF MSD = 1
3811	020102	001405				BEQ	OCTP1	;IF NOT: BR
3812	020104	012704	000001			MOV	#1,R4	
3813	020110	004737	020230			JSR	PC,OCTPG	;PRINT 1
3814	020114	000403				BR	OCTP2	
3815	020116	005004			OCTP1:	CLR	R4	
3816	020120	004737	020230			JSR	PC,OCTPG	;PRINT 0
3817	020124	010304			OCTP2:	MOV	R3,R4	
3818	020126	006004				ROR	R4	
3819	020130	006004				ROR	R4	
3820	020132	006004				ROR	R4	;POSITION DIGIT
3821	020134	006004				ROR	R4	
3822	020136	000304				SWAB	R4	
3823	020140	004737	020230			JSR	PC,OCTPG	;PRINT DIGIT 2
3824	020144	010304				MOV	R3,R4	
3825	020146	006004				ROR	R4	
3826	020150	000304				SWAB	R4	
3827	020152	004737	020230			JSR	PC,OCTPG	;PRINT DIGIT 3
3828	020156	010304				MOV	R3,R4	
3829	020160	006104				ROL	R4	
3830	020162	006104				ROL	R4	
3831	020164	000304				SWAB	R4	
3832	020166	004737	020230			JSR	PC,OCTPG	;PRINT DIGIT 4
3833	020172	010304				MOV	R3,R4	
3834	020174	006004				ROR	R4	
3835	020176	006004				ROR	R4	
3836	020200	006004				ROR	R4	
3837	020202	004737	020230			JSR	PC,OCTPG	
3838	020206	010304				MOV	R3,R4	
3839	020210	004737	020230			JSR	PC,OCTPG	;PRINT DIGIT 5
3840	020214	012737	000240	000610	OCTP3:	MOV	#240,TOB	
3841	020222	004737	020024			JSR	PC,TOG	;PRINT SPACE
3842	020226	000207				RTS	PC	;EXIT
3843								
3844	020230	042704	177770		OCTPG:	BIC	#177770,R4	
3845	020234	001004				BNE	OCTPG0	
3846	020236	005737	020272			TST	OFL	
3847	020242	001001				BNE	OCTPG0	
3848	020244	000207				RTS	PC	
3849								
3850	020246	005237	020272		OCTPG0:	INC	OFL	
3851	020252	052704	000260		OCTPG1:	BIS	#260,R4	
3852	020256	010437	000610			MOV	R4,TOB	
3853	020262	004737	020024			JSR	PC,TOG	
3854	020266	010304				MOV	R3,R4	
3855	020270	000207				RTS	PC	
3856	020272	000000			OFL:	0		;FIRST CHAR FLAG
3857								

```

3859
3860 ;DATA CHARACTER OUTPUT SUBROUTINE*****
3861
3862 020274 012704 000010 DOUT: MOV #10,R4 ;SET NUMBER TO PRINT
3863 020300 110337 000610 MOVB R3,TOB
3864 020304 105777 160266 1$: TSTB @TPS
3865 020310 100375 BPL 1$
3866 020312 132737 000200 000610 BITB #200,TOB
3867 020320 001404 BEQ 2$
3868 020322 012777 000061 160250 MOV #061,@TPB
3869 020330 000403 BR 3$
3870 020332 012777 000060 160240 2$: MOV #060,@TPB
3871 020340 006337 000610 3$: ASL TOB
3872 020344 005304 DEC R4
3873 020346 001356 BNE 1$
3874 020350 000207 RTS PC
3875
3876 020352 013703 000674 DOUTD: MOV TEMP3,R3
3877 020356 000303 SWAB R3
3878 020360 004737 020274 JSR PC,DOUT
3879 020364 013703 000674 MOV TEMP3,R3
3880 020370 004737 020274 JSR PC,DOUT
3881 020374 000207 RTS PC
3882
3883 ;TE16 SERIAL NUMBER PRINT SUBROUTINE*****
3884
3885 020376 010304 SNPT: MOV R3,R4
3886 020400 000304 SWAB R4
3887 020402 006004 ROR R4
3888 020404 006004 ROR R4
3889 020406 006004 ROR R4
3890 020410 006004 ROR R4 ;GET FIRST DIGIT
3891 020412 004737 020454 JSR PC,SNPG ;PRINT
3892 020416 010304 MOV R3,R4
3893 020420 000304 SWAB R4 ;GET SECOND DIGIT
3894 020422 004737 020454 JSR PC,SNPG ;PRINT
3895 020426 010304 MOV R3,R4
3896 020430 006004 ROR R4
3897 020432 006004 ROR R4
3898 020434 006004 ROR R4
3899 020436 006004 ROR R4
3900 020440 004737 020454 JSR PC,SNPG ;PRINT THIRD DIGIT
3901 020444 010304 MOV R3,R4
3902 020446 004737 020454 JSR PC,SNPG ;PRINT FOURTH DIGIT
3903 020452 000207 RTS PC ;EXIT
3904 020454 012737 000260 000610 SNPG: MOV #260,TOB ;SET BASE = 0
3905 020462 042704 177760 BIC #177760,R4 ;MASK DIGIT
3906 020466 050437 000610 BIS R4,TOB ;SET ASCII
3907 020472 004737 020024 JSR PC,TOG ;TYPE DIGIT
3908 020476 000207 RTS PC ;RETURN
  
```



```

3910
3911      ;ROUTINE TO LOAD CONTENTS OF SOFTWARE SWITCH REGISTER.
3912      ;IF A CONTROL G (^G) IS TYPED THE SOFTWARE SWITCH REGISTER IS LOADED
3913 020500 022737 000176 000570 GTSWR:  CMP    #SWREG,SWR      ;BRANCH IF SOFTWARE SWR
3914 020506 001032                BNE    1$          ;NOT INVOKED
3915 020510 004737 020576                JSR    PC,,SAVE   ;SAVE REGISTERS ON THE STACK
3916 020514 012704 027327                MOV    #SMSWR,R4 ;TYPE 'SWR = '
3917 020520 004737 017724                JSR    PC,TTOUT
3918 020524 017703 160040                MOV    @SWR,R3   ;GET CURRENT VALUE
3919 020530 004737 020042                JSR    PC,OCTPE  ;AND TYPE IT
3920 020534 012704 027337                MOV    #SMNEW,R4 ;ASK FOR NEW VALUE
3921 020540 004737 017724                JSR    PC,TTOUT
3922 020544 013705 000570                MOV    SWR,R5   ;NEW VALUE WILL BE RETURNED IN (R5)
3923 020550 012701 000007                MOV    #7,R1    ;LIMIT TO 7 CHARACTERS
3924 020554 012702 177777                MOV    #177777,R2 ;LIMIT RESPONSE TO BETWEEN
3925 020560 012703 000000                MOV    #0,R3    ;0 AND 177777
3926 020564 004737 017402                JSR    PC,TTR   ;GET RESPONSE
3927 020570 004737 020620                JSR    PC,,RESTORE ;RESTORE REGISTERS
3928 020574 000207                RTS    PC       ;RETURN TO CALLER
3929      ;:ROUTINE TO SAVE REGISTERS ON THE STACK
(1) 020576 010546                .SAVE: MOV    %5,-(SP) ;:R5 IS SAVED AT 12(SP)
(1) 020600 010446                MOV    %4,-(SP) ;:R4 IS SAVED AT 10(SP)
(1) 020602 010346                MOV    %3,-(SP) ;:R3 IS SAVED AT 6(SP)
(1) 020604 010246                MOV    %2,-(SP) ;:R2 IS SAVED AT 4(SP)
(1) 020606 010146                MOV    %1,-(SP) ;:R1 IS SAVED AT 2(SP)
(1) 020610 010046                MOV    %0,-(SP) ;:R0 IS SAVED AT (SP)
(1) 020612 016646 000014                MOV    14(SP),-(SP) ;:PUSH RETURN PC ON THE STACK
(1) 020616 000207                RTS    PC       ;:RETURN TO CALLER
3930      ;:ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
(1) 020620 012666 000014                .RESTORE:MOV (SP)+,14(SP) ;:STORE RETURN PC ON STACK
(1) 020624 012600                MOV    (SP)+,%0
(1) 020626 012601                MOV    (SP)+,%1
(1) 020630 012602                MOV    (SP)+,%2
(1) 020632 012603                MOV    (SP)+,%3
(1) 020634 012604                MOV    (SP)+,%4
(1) 020636 012605                MOV    (SP)+,%5
(1) 020640 000207                RTS    PC       ;:RETURN
3931
3932      ;MESSAGE TABLE*****
3933
3934 020642 022445 046524 031460 MSG1:  .ASCII '%TM03-TE16/TU77 CONTROL LOGIC TEST- PART I (CZTEADO)';++B
020650 052055 030505 027466
020656 052524 033467 041440
020664 047117 051124 046117
020672 046040 043517 041511
020700 052040 051505 026524
020706 050040 051101 020124
020714 020111 041450 052132
020722 040505 030104 051
3935 020727 045 025052 040452 .ASCII '/%***ASSURE TAPE IS AT BOT***/'
020734 051523 051125 020105
020742 040524 042520 044440
020750 020123 052101 041040
020756 052117 025052 052

```

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

	021434	051523	052040	051505	
	021442	035524			
3947	021444	020045	047105	042524	.ASCII /% ENTER EXPT SLAVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
	021452	020122	054105	052120	
	021460	051440	040514	042526	
	021466	047040	046525	042502	
	021474	026122	040440	046114	
	021502	047440	044124	051105	
	021510	020123	044123	052517	
	021516	042114	041040	020105	
	021524	047516	026516	054105	
	021532	051511	040524	052116	
	021540	021456			
3948	021542	047045	047117	042455	MSG9: .ASCII /%NON-EXIST SLAVE #/
	021550	044530	052123	051440	
	021556	040514	042526	021440	
3949	021564	051045	040505	020104	MSG10: .ASCII /%READ CONT BUS PAR #/
	021572	047503	052116	041040	
	021600	051525	050040	051101	
	021606	021440			
3950	021610	053445	044522	042524	MSG11: .ASCII /%WRITE CONT BUS PAR #/
	021616	041440	047117	020124	
	021624	052502	020123	040520	
	021632	020122	043		
3951	021635	040	054105	052120	MSG12: .ASCII / EXPT #/
	021642	021440			
3952	021644	051040	053103	020104	MSG13: .ASCII / RCVD #/
	021652	043			
3953	021653	045	051115	041040	MSG14: .ASCII /%MR BITS 4-0#/
	021660	052111	020123	026464	
	021666	021460			
3954	021670	046445	020122	044502	MSG15: .ASCII /%MR BITS 15-7#/
	021676	051524	030440	026465	
	021704	021467			
3955	021706	044445	042524	035122	MSG16: .ASCII /%ITER: #/
	021714	021440			
3956	021716	052045	020103	044502	MSG18: .ASCII /%TC BITS 12-0 #/
	021724	051524	030440	026462	
	021732	020060	043		
3957	021735	045	041506	041040	MSG19: .ASCII /%FC BITS 15-0 #/
	021742	052111	020123	032461	
	021750	030055	021440		
3958	021754	043045	047125	041440	MSG20: .ASCII /%FUN CODE BITS 5-1 OF C1 #/
	021762	042117	020105	044502	
	021770	051524	032440	030455	
	021776	047440	020106	030503	
	022004	021440			
3959	022006	043445	020117	044502	MSG21: .ASCII /%GO BIT NOT CORRECT AT START #/
	022014	020124	047516	020124	
	022022	047503	051122	041505	
	022030	020124	052101	051440	
	022036	040524	052122	021440	
3960	022044	043445	020117	044502	MSG22: .ASCII /%GO BIT NOT SET #/
	022052	020124	047516	020124	
	022060	042523	020124	043	
3961	022065	045	047507	041040	MSG23: .ASCII /%GO BIT NOT RESET BY INIT #/

	022072	052111	047040	052117			
	022100	051040	051505	052105			
	022106	041040	020131	047111			
	022114	052111	021440				
3962	022120	042045	054522	047040	MSG24:	.ASCII	/%DRY NOT SET BY INIT #/
	022126	052117	051440	052105			
	022134	041040	020131	047111			
	022142	052111	021440				
3963	022146	042045	054522	047040	MSG25:	.ASCII	/%DRY NOT RESET BY GO=1#/
	022154	052117	051040	051505			
	022162	052105	041040	020131			
	022170	047507	030475	043			
3964	022175	045	051104	020131	MSG25A:	.ASCII	/%DRY NOT SET BY GO=0#/
	022202	047516	020124	042523			
	022210	020124	054502	043440			
	022216	036517	021460				
3965	022222	047045	020117	047111	MSG26:	.ASCII	/%NO INTERRUPT RETURNED#/
	022230	042524	051122	050125			
	022236	020124	042522	052524			
	022244	047122	042105	043			
3966	022251	045	040502	020104	MSG27:	.ASCII	/%BAD STATUS#/
	022256	052123	052101	051525			
	022264	043					
3967	022265	040	047123	020072	MSG30:	.ASCII	/ SN: #/
	022272	043					
3968	022273	045	051105	020122	MSG31:	.ASCII	/%ERR NOT SET #/
	022300	047516	020124	042523			
	022306	020124	043				
3969	022311	045	052101	020101	MSG32:	.ASCII	/%ATA NOT SET #/
	022316	047516	020124	042523			
	022324	020124	043				
3970	022327	045	051501	041040	MSG33:	.ASCII	/%AS BIT NOT SET #/
	022334	052111	047040	052117			
	022342	051440	052105	021440			
3971	022350	051445	020103	047516	MSG34:	.ASCII	/%SC NOT SET #/
	022356	020124	042523	020124			
	022364	043					
3972	022365	045	051124	020105	MSG35:	.ASCII	/%TRE NOT SET #/
	022372	047516	020124	042523			
	022400	020124	043				
3973	022403	045	046123	020101	MSG36:	.ASCII	/%SLA NOT SET #/
	022410	047516	020124	042523			
	022416	020124	043				
3974	022421	045	051523	020103	MSG37:	.ASCII	/%SSC NOT SET #/
	022426	047516	020124	042523			
	022434	020124	043				
3975	022437	040	020077	043	MSG40:	.ASCII	/ ? #/
3976	022443	045	042445	042116	MSG41:	.ASCII	/%END OF PASS #/
	022450	047440	020106	040520			
	022456	051523	021440				
3977	022462	042045	040505	020104	MSG42:	.ASCII	/%DEAD TRACK #/
	022470	051124	041501	020113			
	022476	043					
3978	022477	045	046445	047101	MSG43:	.ASCII	/%MANUAL TESTS (14-17) INHIBITED: HALT%/
	022504	040525	020114	042524			
	022512	052123	020123	030450			

	022520	026464	033461	020051	
	022526	047111	044510	044502	
	022534	042524	035104	044040	
	022542	046101	022524		
3979	022546	042522	042523	042514	.ASCII /RESELECT AND PRESS CONTINUE%/
	022554	052103	040440	042116	
	022562	050040	042522	051523	
	022570	041440	047117	044524	
	022576	052516	022505	043	
3980	022603	045	042522	044507	MSG44: .ASCII /%REGISTER START: #/
	022610	052123	051105	051440	
	022616	040524	052122	020072	
	022624	043			
3981	022625	045	042526	052103	MSG45: .ASCII /%VECTOR ADDRESS: #/
	022632	051117	040440	042104	
	022640	042522	051523	020072	
	022646	043			
3982	022647	045	051503	020061	MSG46: .ASCII /%CS1 WC BA FC CS2 DS ER AS/
	022654	020040	053440	020103	
	022662	020040	020040	040502	
	022670	020040	020040	043040	
	022676	020103	020040	020040	
	022704	051503	020062	020040	
	022712	042040	020123	020040	
	022720	020040	051105	020040	
	022726	020040	040440	123	
3983	022733	040	020040	020040	.ASCII / MR TC%/
	022740	051115	020040	020040	
	022746	052040	022503	043	
3984	022753	045	047516	020124	MSG47: .ASCII /%NOT RESET BY DRIVE CLEAR%/
	022760	042522	042523	020124	
	022766	054502	042040	044522	
	022774	042526	041440	042514	
	023002	051101	043		
3985	023005	045	046101	044120	MSG50: .ASCII /%ALPHA NOT SET%/
	023012	020101	047516	020124	
	023020	042523	021524		
3986	023024	052445	042516	050130	MSG51: .ASCII /%UNEXPECTED ERROR BITS%/
	023032	041505	042524	020104	
	023040	051105	047522	020122	
	023046	044502	051524	043	
3987	023053	045	040502	020104	MSG53: .ASCII /%BAD LRC #/
	023060	051114	020103	043	
3988	023065	045	040502	020104	MSG54: .ASCII /%BAD CK #/
	023072	045503	021440		
3989	023076	051445	052105	050125	MSG55: .ASCII /%SETUP ERROR: CHECK WRAP 0 WITH CONTROL LOGIC TEST II TEST 7%/
	023104	042440	051122	051117	
	023112	020072	044103	041505	
	023120	020113	051127	050101	
	023126	030040	053440	052111	
	023134	020110	047503	052116	
	023142	047522	020114	047514	
	023150	044507	020103	042524	
	023156	052123	044440	020111	
	023164	042524	052123	033440	
	023172	043			

3990	023173	045	052123	052101	MSG56:	.ASCII	/%STATIC TESTS ONLY: #/
	023200	041511	052040	051505			
	023206	051524	047440	046116			
	023214	035131	021440				
3991	023220	052045	047515	020063	MSG57:	.ASCII	/%TM03 DRIVE: #/
	023226	051104	053111	035105			
	023234	021440					
3992	023236	044445	020123	047503	MS57A:	.ASCII	/%IS CONTROLLER JUMPERED IN NON-STANDARD MODE,/<15><12>
	023244	052116	047522	046114			
	023252	051105	045040	046525			
	023260	042520	042522	020104			
	023266	047111	047040	047117			
	023274	051455	040524	042116			
	023302	051101	020104	047515			
	023310	042504	006454	012			
3993	023315	124	050131	020105		.ASCII	/TYPE 2 FOR NON=STANDARD OR CR FOR STANDARD ? #/
	023322	020062	047506	020122			
	023330	047516	036516	052123			
	023336	047101	040504	042122			
	023344	047440	020122	051103			
	023352	043040	051117	051440			
	023360	040524	042116	051101			
	023366	020104	020077	020040			
	023374	020040	043				
3994	023377	045	042524	033061	MSG58:	.ASCII	'%TE16/TU77 SLAVE: #' ; ++B
	023404	052057	033525	020067			
	023412	046123	053101	035105			
	023420	021440					
3995	023422	021445			MSG59:	.ASCII	/##/
3996	023424	021534			MSG60:	.ASCII	/\##/
3997	023426	051045	046505	053117	MSG62:	.ASCII	/%REMOVE TMDP FROM SLAVE TO BE TESTED##/
	023434	020105	046524	050104			
	023442	043040	047522	020115			
	023450	046123	053101	020105			
	023456	047524	041040	020105			
	023464	042524	052123	042105			
	023472	021445					
3998	023474	044045	051101	053504	MSG63:	.ASCII	/%HARDWARE SWR IN USE##/
	023502	051101	020105	053523			
	023510	020122	047111	052440			
	023516	042523	021445				
3999	023522	051445	040514	042526	MSG64:	.ASCII	/%SLAVE TYPE: #/ ; ++B
	023530	052040	050131	035105			
	023536	021440					
4000	023540	052524	033467	043	MSG65:	.ASCII	/TU77##/ ; ++B
4001	023545	124	030505	021466	MSG66:	.ASCII	/TE16##/ ; ++B
4002	023552	051445	040514	042526	MSG67:	.ASCII	/%SLAVE TYPE (0=TE16,1=TU77): #/ ; ++B
	023560	052040	050131	020105			
	023566	030050	052075	030505			
	023574	026066	036461	052524			
	023602	033467	035051	021440			
4003	023610	022445	047111	047503	MSG68:	.ASCII	/%INCORRECT SLAVE TYPE!!! PROGRAM ABORTED##/ ; ++B
	023616	051122	041505	020124			
	023624	046123	053101	020105			
	023632	054524	042520	020441			
	023640	020041	051120	043517			

4004	023646	040522	020115	041101		
	023654	051117	042524	021504		
	023662	046111	042514	040507	MSG69:	.ASCII /ILLEGAL#/ ;++B
4005	023670	021514				;TEST HEADER*****
4006						
4007	023672	022445	047514	044507	MSLT1:	.ASCII /%%LOGIC TEST 1: DRIVE ADDRESSING (M8909 RH)#/
	023700	020103	042524	052123		
	023706	030440	020072	051104		
	023714	053111	020105	042101		
	023722	051104	051505	044523		
	023730	043516	024040	034115		
	023736	030071	020071	044122		
	023744	021451				
4008	023746	022445	047514	044507	MSLT2:	.ASCII /%%LOGIC TEST 2: REGISTER ADDRESSING (M8909 RH)#/
	023754	020103	042524	052123		
	023762	031040	020072	042522		
	023770	044507	052123	051105		
	023776	040440	042104	042522		
	024004	051523	047111	020107		
	024012	046450	034470	034460		
	024020	051040	024510	043		
4009	024025	045	046045	043517	MSLT3:	.ASCII /%%LOGIC TEST 3: CONTROL BUS TEST (RH M8905-YB M8909)#/
	024032	041511	052040	051505		
	024040	020124	035063	041440		
	024046	047117	051124	046117		
	024054	041040	051525	052040		
	024062	051505	020124	051050		
	024070	020110	034115	030071		
	024076	026465	041131	046440		
	024104	034470	034460	021451		
4010	024112	022445	047514	044507	MSLT4:	.ASCII /%%LOGIC TEST 4: SLAVE ADDRESSING (M8905-YB M8933)#/
	024120	020103	042524	052123		
	024126	032040	020072	046123		
	024134	053101	020105	042101		
	024142	051104	051505	044523		
	024150	043516	024040	034115		
	024156	030071	026465	041131		
	024164	046440	034470	031463		
	024172	021451				
4011	024174	022445	047514	044507	MSLT5:	.ASCII /%%LOGIC TEST 5: MR BIT TEST (M8905-YB)#/
	024202	020103	042524	052123		
	024210	032440	020072	051115		
	024216	041040	052111	052040		
	024224	051505	020124	046450		
	024232	034470	032460	054455		
	024240	024502	043			
4012	024243	045	046045	043517	MSLT6:	.ASCII /%%LOGIC TEST 6: TC BIT TEST (M8905-YB)#/
	024250	041511	052040	051505		
	024256	020124	035066	052040		
	024264	020103	044502	020124		
	024272	042524	052123	024040		
	024300	034115	030071	026465		
	024306	041131	021451			
4013	024312	022445	047514	044507	MSLT7:	.ASCII /%%LOGIC TEST 7: FC BIT TEST (M8905-YB)#/
	024320	020103	042524	052123		

	024326	033440	020072	041506	
	024334	041040	052111	052040	
	024342	051505	020124	046450	
	024350	034470	032460	054455	
	024356	024502	043		
4014	024361	045	046045	043517	MSLT10: .ASCII /%%LOGIC TEST 10: FUNCTION BIT TEST (M8905-YB)#/
	024366	041511	052040	051505	
	024374	020124	030061	020072	
	024402	052506	041516	044524	
	024410	047117	041040	052111	
	024416	052040	051505	020124	
	024424	046450	034470	032460	
	024432	054455	024502	043	
4015	024437	045	046045	043517	MSLT11: .ASCII /%%LOGIC TEST 11: GO BIT TEST (M8909)#/
	024444	041511	052040	051505	
	024452	020124	030461	020072	
	024460	047507	041040	052111	
	024466	052040	051505	020124	
	024474	046450	034470	034460	
	024502	021451			
4016	024504	022445	047514	044507	MSLT12: .ASCII /%%LOGIC TEST 12: DRIVE READY BIT (M8909)#/
	024512	020103	042524	052123	
	024520	030440	035062	042040	
	024526	044522	042526	051040	
	024534	040505	054504	041040	
	024542	052111	024040	034115	
	024550	030071	024471	043	
4017	024555	045	046045	043517	MSLT13: .ASCII /%%LOGIC TEST 13: INTERRUPT TEST (RH)#/
	024562	041511	052040	051505	
	024570	020124	031461	020072	
	024576	047111	042524	051122	
	024604	050125	020124	042524	
	024612	052123	024040	044122	
	024620	021451			
4018	024622	022445	047514	044507	MSLT14: .ASCII /%%LOGIC TEST 14: STATUS AT BOT,ON-LINE,WRITE PROTECTED (NO WRITE RING)#
	024630	020103	042524	052123	
	024636	030440	035064	051440	
	024644	040524	052524	020123	
	024652	052101	041040	052117	
	024660	047454	026516	044514	
	024666	042516	053454	044522	
	024674	042524	050040	047522	
	024702	042524	052103	042105	
	024710	024040	047516	053440	
	024716	044522	042524	051040	
	024724	047111	024507	043	
4019	024731	045	046045	043517	MSLT15: .ASCII /%%LOGIC TEST 15: STATUS AT BOT,OFF-LINE,WRITE PROTECTED#/
	024736	041511	052040	051505	
	024744	020124	032461	020072	
	024752	052123	052101	051525	
	024760	040440	020124	047502	
	024766	026124	043117	026506	
	024774	044514	042516	053454	
	025002	044522	042524	050040	
	025010	047522	042524	052103	
	025016	042105	043		

4020	025021	045	046045	043517	MSLT16: .ASCII /%%LOGIC TEST 16: STATUS AT EOT,ON-LINE,WRITE PROTECTED#/
	025026	041511	052040	051505	
	025034	020124	033061	020072	
	025042	052123	052101	051525	
	025050	040440	020124	047505	
	025056	026124	047117	046055	
	025064	047111	026105	051127	
	025072	052111	020105	051120	
	025100	052117	041505	042524	
	025106	021504			
4021	025110	022445	047514	044507	MSLT17: .ASCII /%%LOGIC TEST 17: STATUS AT ON-LINE,WRITE ENABLED#/
	025116	020103	042524	052123	
	025124	030440	035067	051440	
	025132	040524	052524	020123	
	025140	052101	047440	026516	
	025146	044514	042516	053454	
	025154	044522	042524	042440	
	025162	040516	046102	042105	
	025170	043			
4022	025171	045	046045	043517	MSLT20: .ASCII /%%LOGIC TEST 20: ILLEGAL FUNCTION TEST (M8909)#/
	025176	041511	052040	051505	
	025204	020124	030062	020072	
	025212	046111	042514	040507	
	025220	020114	052506	041516	
	025226	044524	047117	052040	
	025234	051505	020124	046450	
	025242	034470	034460	021451	
4023	025250	022445	047514	044507	MSLT21: .ASCII /%%LOGIC TEST 21: RMR(M8909)#/
	025256	020103	042524	052123	
	025264	031040	035061	051040	
	025272	051115	046450	034470	
	025300	034460	021451		
4024	025304	022445	047514	044507	MSLT22: .ASCII /%%LOGIC TEST 22: CPAR(M8909)#/
	025312	020103	042524	052123	
	025320	031040	035062	041440	
	025326	040520	024122	034115	
	025334	030071	024471	043	
4025	025341	045	046045	043517	MSLT23: .ASCII /%%LOGIC TEST 23: FMT(M8905-YB M8906)#/
	025346	041511	052040	051505	
	025354	020124	031462	020072	
	025362	046506	024124	034115	
	025370	030071	026465	041131	
	025376	046440	034470	033060	
	025404	021451			
4026	025406	022445	047514	044507	MSLT24: .ASCII /%%LOGIC TEST 24: DPAR(M8906 RH)#/
	025414	020103	042524	052123	
	025422	031040	035064	042040	
	025430	040520	024122	034115	
	025436	030071	020066	044122	
	025444	021451			
4027	025446	022445	047514	044507	MSLT25: .ASCII /%%LOGIC TEST 25: NEF(M8909)#/
	025454	020103	042524	052123	
	025462	031040	035065	047040	
	025470	043105	046450	034470	
	025476	034460	021451		
4028	025502	022445	047514	044507	MSLT26: .ASCII /%%LOGIC TEST 26: FCE(M8909)#/

	025510	020103	042524	052123	
	025516	031040	035066	043040	
	025524	042503	046450	034470	
	025532	034460	021451		
4029	025536	022445	047514	044507	MSLT27: .ASCII /%%LOGIC TEST 27: ILR(M8909)#/
	025544	020103	042524	052123	
	025552	031040	035067	044440	
	025560	051114	046450	034470	
	025566	034460	021451		
4030	025572	022445	047514	044507	MSLT30: .ASCII /%%LOGIC TEST 30:DTE(M8906 RH)#/
	025600	020103	042524	052123	
	025606	031440	035060	052104	
	025614	024105	034115	030071	
	025622	020066	044122	021451	
4031	025630	022445	047514	044507	MSLT31: .ASCII /%%LOGIC TEST 31: OPI(M8933)#/
	025636	020103	042524	052123	
	025644	031440	035061	047440	
	025652	044520	046450	034470	
	025660	031463	021451		
4032	025664	022445	047514	044507	MSLT32: .ASCII /%%LOGIC TEST 32: UNS(M8909)#/
	025672	020103	042524	052123	
	025700	031440	035062	052440	
	025706	051516	046450	034470	
	025714	034460	021451		
4033	025720	022445	047514	044507	MSLT33: .ASCII /%%LOGIC TEST 33: PIP(M8909)#/
	025726	020103	042524	052123	
	025734	031440	035063	050040	
	025742	050111	046450	034470	
	025750	034460	021451		
4034	025754	022445	047514	044507	MSLT34: .ASCII /%%LOGIC TEST 34: PES(M8931)#/
	025762	020103	042524	052123	
	025770	031440	035064	050040	
	025776	051505	046450	034470	
	026004	030463	021451		
4035	026010	022445	047514	044507	MSLT35: .ASCII /%%LOGIC TEST 35: SAC(M8933 M8905-YB)#/
	026016	020103	042524	052123	
	026024	031440	035065	051440	
	026032	041501	046450	034470	
	026040	031463	046440	034470	
	026046	032460	054455	024502	
	026054	043			
4036	026055	045	046045	043517	MSLT36: .ASCII /%%LOGIC TEST 36: FCS(M8933 M8905-YB)#/
	026062	041511	052040	051505	
	026070	020124	033063	020072	
	026076	041506	024123	034115	
	026104	031471	020063	034115	
	026112	030071	026465	041131	
	026120	021451			
4037	026122	022445	047514	044507	MSLT37: .ASCII /%%LOGIC TEST 37: ACCL(M8933 M8905-YB)#/
	026130	020103	042524	052123	
	026136	031440	035067	040440	
	026144	041503	024114	034115	
	026152	031471	020063	034115	
	026160	030071	026465	041131	
	026166	021451			
4038	026170	022445	047514	044507	MSLT40: .ASCII /%%LOGIC TEST 40: PE TAPE MARK(M8932)#/

	026176	020103	042524	052123	
	026204	032040	035060	050040	
	026212	020105	040524	042520	
	026220	046440	051101	024113	
	026226	034115	031471	024462	
	026234	043			
4039	026235	045	046045	043517	MSLT41: .ASCII /%%LOGIC TEST 41: NRZ TAPE MARK (M8934)#/
	026242	041511	052040	051505	
	026250	020124	030464	020072	
	026256	051116	020132	040524	
	026264	042520	046440	051101	
	026272	020113	046450	034470	
	026300	032063	021451		
4040	026304	022445	047514	044507	MSLT42: .ASCII /%%LOGIC TEST 42: CRC(M8934)#/
	026312	020103	042524	052123	
	026320	032040	035062	041440	
	026326	041522	046450	034470	
	026334	032063	021451		
4041	026340	022445	047514	044507	MSLT43: .ASCII /%%LOGIC TEST 43: LRC(M8934)#/
	026346	020103	042524	052123	
	026354	032040	035063	046040	
	026362	041522	046450	034470	
	026370	032063	021451		
4042	026374	022445	047514	044507	MSLT44: .ASCII /%%LOGIC TEST 44: CORRECTABLE DATA (M8932 M8901)#/
	026402	020103	042524	052123	
	026410	032040	035064	041440	
	026416	051117	042522	052103	
	026424	041101	042514	042040	
	026432	052101	020101	046450	
	026440	034470	031063	046440	
	026446	034470	030460	021451	
4043	026454	022445	047514	044507	MSLT45: .ASCII /%%LOGIC TEST 45: INCORRECTABLE DATA (M8932 M8934)#/
	026462	020103	042524	052123	
	026470	032040	035065	044440	
	026476	041516	051117	042522	
	026504	052103	041101	042514	
	026512	042040	052101	020101	
	026520	046450	034470	031063	
	026526	046440	034470	032063	
	026534	021451			
4044	026536	022445	047514	044507	MSLT46: .ASCII /%%LOGIC TEST 46: PEF(M8932)#/
	026544	020103	042524	052123	
	026552	032040	035066	050040	
	026560	043105	046450	034470	
	026566	031063	021451		
4045	026572	022445	047514	044507	MSLT47: .ASCII /%%LOGIC TEST 47: FC OVERFLOW (M8905-YB)#/
	026600	020103	042524	052123	
	026606	032040	035067	043040	
	026614	020103	053117	051105	
	026622	046106	053517	024040	
	026630	034115	030071	026465	
	026636	041131	021451		
4046	026642	022445	047514	044507	MSLT50: .ASCII /%%LOGIC TEST 50: NEF WHEN WRITE PE ON NRZ SLAVE#/
	026650	020103	042524	052123	
	026656	032440	035060	047040	
	026664	043105	053440	042510	

407 026672 020116 051127 052111
026700 020105 042520 047440
026706 020116 051116 020132
026714 046123 053101 021505
026722 022445 047514 044507
026730 020103 042524 052123
026736 032440 035061 047040
026744 043105 053440 042510
026752 020116 051127 052111
026760 020105 051116 020132
026766 047117 050040 020105
026774 046123 053101 021505

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


```
4049
4050 ;MANUAL INSTRUCTION*****
4051
4052 027002 052045 050131 020105 MMSG0: .ASCII /%TYPE CR WHEN READY;#/
      027010 051103 053440 042510
      027016 020116 042522 042101
      027024 035531 043
4053 027027 045 046445 052517 MMSG1: .ASCII /%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE:#/
      027034 052116 052040 050101
      027042 020105 044527 044124
      027050 047040 020117 051127
      027056 052111 020105 044522
      027064 043516 020054 047514
      027072 042101 052040 020117
      027100 047502 026124 051440
      027106 052105 052040 020117
      027114 047117 046040 047111
      027122 035105 043
4054 027125 045 042523 020124 MMSG2: .ASCII /%SET TO OFFLINE:#/
      027132 047524 047440 043106
      027140 044514 042516 021472
4055 027146 046445 053117 020105 MMSG3: .ASCII /%MOVE FORWARD TO EOT, ONLINE:#/
      027154 047506 053522 051101
      027162 020104 047524 042440
      027170 052117 020054 047117
      027176 044514 042516 021472
4056 027204 047445 043106 046040 MMSG4: .ASCII /%OFF LINE REVERSE PAST EOT, INSERT WRITE RING, ON LINE#/
      027212 047111 020105 042522
      027220 042526 051522 020105
      027226 040520 052123 042440
      027234 052117 020054 047111
      027242 042523 052122 053440
      027250 044522 042524 051040
      027256 047111 026107 047440
      027264 020116 044514 042516
      027272 043
4057 027273 045 046445 053117 MMSG5: .ASCII /%MOVE TAPE TO BOT; ON LINE#/
      027300 020105 040524 042520
      027306 052040 020117 047502
      027314 035524 047440 020116
      027322 044514 042516 043
```

```

4059
4060                                     ;TAG MESSAGE
4061
4062 027327      045 053523 020122 SMSWR: .ASCII /%SWR = #/
      027334      020075      043
4063 027337      040 042516 020127 SMNEW: .ASCII / NEW = #/
      027344      020075      043
4064 027347      045 046123 020101 TMS1:  .ASCII /%SLA #/
      027354      043
4065 027355      045 047502 020124 TMS2:  .ASCII /%BOT #/
      027362      043
4066 027363      045 046524 021440 TMS3:  .ASCII /%TM #/
4067 027370      044445 041104 021440 TMS4:  .ASCII /%IDB #/
4068 027376      051445 053504 020116 TMS5:  .ASCII /%SDWN #/
      027404      043
4069 027405      045 042520 020123 TMS6:  .ASCII /%PES #/
      027412      043
4070 027413      045 051523 020103 TMS7:  .ASCII /%SSC #/
      027420      043
4071 027421      045 051104 020131 TMS8:  .ASCII /%DRY #/
      027426      043
4072 027427      045 050104 020122 TMS9:  .ASCII /%DPR #/
      027434      043
4073 027435      045 052116 020114 TMS10: .ASCII /%NTL #/
      027442      043
4074 027443      045 047505 020124 TMS11: .ASCII /%EOT #/
      027450      043
4075 027451      045 051127 020114 TMS12: .ASCII /%WRL #/
      027456      043
4076 027457      045 047515 020114 TMS13: .ASCII /%MOL #/
      027464      043
4077 027465      045 044520 020120 TMS14: .ASCII /%PIP #/
      027472      043
4078 027473      045 051105 020122 TMS15: .ASCII /%ERR #/
      027500      043
4079 027501      045 052101 020101 TMS16: .ASCII /%ATA #/
      027506      043
4080 027507      045 046111 020106 TMS17: .ASCII /%ILF #/
      027514      043
4081 027515      045 046111 020122 TMS18: .ASCII /%ILR #/
      027522      043
4082 027523      045 046522 020122 TMS19: .ASCII /%RMR #/
      027530      043
4083 027531      045 050103 051101 TMS20: .ASCII /%CPAR #/
      027536      021440
4084 027540      043045 052115 021440 TMS21: .ASCII /%FMT #/
4085 027546      042045 040520 020122 TMS22: .ASCII /%DPAR #/
      027554      043
4086 027555      045 047111 020103 TMS23: .ASCII /%INC #/
      027562      043
4087 027563      045 050126 020105 TMS24: .ASCII /%VPE #/
      027570      043
4088 027571      045 042520 020106 TMS25: .ASCII /%PEF #/
      027576      043
4089 027577      045 051114 020103 TMS26: .ASCII /%LRC #/
      027604      043
  
```


(1)	030100	177777	-1
(1)	030102	177777	-1
(1)	030104	177777	-1
(1)	030106	177777	-1
(1)	030110	177777	-1
(1)	030112	177777	-1
(1)	030114	177777	-1
(1)	030116	177777	-1
(1)	030120	177777	-1
(1)	030122	177777	-1
(1)	030124	177777	-1
(1)	030126	177777	-1
(1)	030130	177777	-1
(1)	030132	177777	-1
(1)	030134	177777	-1
(1)	030136	177777	-1
(1)	030140	177777	-1
(1)	030142	177777	-1
(1)	030144	177777	-1
(1)	030146	177777	-1
(1)	030150	177777	-1
(1)	030152	177777	-1
(1)	030154	177777	-1
(1)	030156	177777	-1
(1)	030160	177777	-1
(1)	030162	177777	-1
(1)	030164	177777	-1
(1)	030166	177777	-1
(1)	030170	177777	-1
(1)	030172	177777	-1
(1)	030174	177777	-1
(1)	030176	177777	-1
(1)	030200	177777	-1
(1)	030202	177777	-1
(1)	030204	177777	-1
(1)	030206	177777	-1
(1)	030210	177777	-1
(1)	030212	177777	-1
(1)	030214	177777	-1
(1)	030216	177777	-1
(1)	030220	177777	-1
(1)	030222	177777	-1
(1)	030224	177777	-1
(1)	030226	177777	-1
(1)	030230	177777	-1
(1)	030232	177777	-1
(1)	030234	177777	-1
(1)	030236	177777	-1

4115
4116
4117
4118
4119
4121
(1)
(1)

030240 000100
030240 000000
030242 000000
030244 000000

RDATA:

:READ BUFFER
0
0
0

(1)	030246	000000	0
(1)	030250	000000	0
(1)	030252	000000	0
(1)	030254	000000	0
(1)	030256	000000	0
(1)	030260	000000	0
(1)	030262	000000	0
(1)	030264	000000	0
(1)	030266	000000	0
(1)	030270	000000	0
(1)	030272	000000	0
(1)	030274	000000	0
(1)	030276	000000	0
(1)	030300	000000	0
(1)	030302	000000	0
(1)	030304	000000	0
(1)	030306	000000	0
(1)	030310	000000	0
(1)	030312	000000	0
(1)	030314	000000	0
(1)	030316	000000	0
(1)	030320	000000	0
(1)	030322	000000	0
(1)	030324	000000	0
(1)	030326	000000	0
(1)	030330	000000	0
(1)	030332	000000	0
(1)	030334	000000	0
(1)	030336	000000	0
(1)	030340	000000	0
(1)	030342	000000	0
(1)	030344	000000	0
(1)	030346	000000	0
(1)	030350	000000	0
(1)	030352	000000	0
(1)	030354	000000	0
(1)	030356	000000	0
(1)	030360	000000	0
(1)	030362	000000	0
(1)	030364	000000	0
(1)	030366	000000	0
(1)	030370	000000	0
(1)	030372	000000	0
(1)	030374	000000	0
(1)	030376	000000	0
(1)	030400	000000	0
(1)	030402	000000	0
(1)	030404	000000	0
(1)	030406	000000	0
(1)	030410	000000	0
(1)	030412	000000	0
(1)	030414	000000	0
(1)	030416	000000	0
(1)	030420	000000	0
(1)	030422	000000	0
(1)	030424	000000	0

(1) 030426 000000 0
(1) 030430 000000 0
(1) 030432 000000 0
(1) 030434 000000 0
(1) 030436 000000 0

;WRAP AROUND MESSAGES*****

4122
4123
4124
4125 030440 042045 052101 020101 WMSG27: .ASCII /%DATA PAT:#/

030446	040520	035124	043
--------	--------	--------	-----

4126 030453 045 047505 020122 WMSG31: .ASCII /%EOR CLEAR DID NOT CLEAR GO%/

030460	046103	040505	020122
030466	044504	020104	047516
030474	020124	046103	040505
030502	020122	047507	021445

4127
4128
4129 030510 000000 PRE: .EVEN
4132 030512 000000
(1) 030514 000000
(1) 030516 000000
(1) 030520 000000
(1) 030522 000000
(1) 030524 000000
(1) 030526 000000
(1) 030530 000000
(1) 030532 000000
(1) 030534 000000
(1) 030536 000000
(1) 030540 000000
(1) 030542 000000
(1) 030544 000000
(1) 030546 000000
(1) 030550 000000
(1) 030552 000000
(1) 030554 000000
(1) 030556 000000
(1) 030560 000000
(1) 030562 000000
(1) 030564 000000
(1) 030566 000000
(1) 030570 000000
(1) 030572 000000
(1) 030574 000000
(1) 030576 000000
(1) 030600 000000
(1) 030602 000000
(1) 030604 000000
(1) 030606 000000
(1) 030610 000000
(1) 030612 000000
(1) 030614 000000
(1) 030616 000000
(1) 030620 000000
(1) 030622 000000
(1) 030624 000000

(1)	030626	000000		0
(1)	030630	000000		0
4133	030632	000000	POST:	0
4136	030634	000000		0
(1)	030636	000000		0
(1)	030640	000000		0
(1)	030642	000000		0
(1)	030644	000000		0
(1)	030646	000000		0
(1)	030650	000000		0
(1)	030652	000000		0
(1)	030654	000000		0
(1)	030656	000000		0
(1)	030660	000000		0
(1)	030662	000000		0
(1)	030664	000000		0
(1)	030666	000000		0
(1)	030670	000000		0
(1)	030672	000000		0
(1)	030674	000000		0
(1)	030676	000000		0
(1)	030700	000000		0
(1)	030702	000000		0
(1)	030704	000000		0
(1)	030706	000000		0
(1)	030710	000000		0
(1)	030712	000000		0
(1)	030714	000000		0
(1)	030716	000000		0
(1)	030720	000000		0
(1)	030722	000000		0
(1)	030724	000000		0
(1)	030726	000000		0
(1)	030730	000000		0
(1)	030732	000000		0
(1)	030734	000000		0
(1)	030736	000000		0
(1)	030740	000000		0
(1)	030742	000000		0
(1)	030744	000000		0
(1)	030746	000000		0
(1)	030750	000000		0
(1)	030752	000000		0
4137	030754	000000	WBUFF:	0
4138		031366		.=.+410
4139	031366	000000	RBUFF:	0
4140				
4141		000001		.END

CZTEADO TM03-TE16/TU77 CTL I CZTEAD.P11 06-JUL-83 14:40		MACY11 30(1046) 06-JUL-83 21:01		E 10 PAGE 87-1 CROSS REFERENCE TABLE -- USER SYMBOLS											SEQ 0121
ER	000524	1346#	1804	1836	2261	2290	2319	2344	2391	2404	2426	2462	2468	2474	
		2493	2518	2544	2550	2579	2582	2602	2605	2727	2901	2906	2913	2951	
		2972	2979	3009	3022	3053	3067	3099	3111	3145	3151	3213	3245	3379	
		3407													
ERADD	000666	1404#	1783*	1813*	1816*	1819*	1845*	1850*	1929*	1963*	1967*	1993*	2018*	2046*	
		2074*	2080*	2086*	2112*	2116*	2120*	2139*	2166*	2189*	2211*	2237*	2264*	2293*	
		2322*	2347*	2398*	2430*	2470*	2496*	2509*	2566*	2591*	2655	2679	2690	2730*	
		2757*	2766*	2789*	2816*	2837*	2877*	2898*	2903*	2908*	2974*	3011*	3020*	3055*	
		3064*	3101*	3109*	3147*	3189*	3305*	3316	3497						
ERRF	000722	1418#													
EXFL	000712	1414#	1780*	1782*	1812*	1815*	1818*	1925*	1927*	2265*	2294*	2323*	2348*	2399*	
		2431*	2471*	2495*	2520*	2546*	2731*	2758*	2774*	2781*	2799*	2807*	2821*	2828*	
		2842*	2855*	2878*	2899*	2904*	2909*	2975*	3012*	3056*	3102*	3148*	3190*	3216*	
		3248*	3304*	3322	3325	3359*									
FC	000516	1343#	1835*	1838	1847	2008*	2009	2063*	2103*	2288*	2316*	2367*	2422*	2451*	
		2516*	2525*	2574*	2753*	2824*	2890*	2942*	2963*	3001*	3039*	3043	3085*	3089	
		3130*	3133	3170*	3185	3205*	3237*	3373							
FUN	000746	1428#													
GTSWR	020500	1681	3678	3913#											
HDRFL	000614	1383#	1702*	3288	3311	3315*	3424	3466	3474*	3492	3496*				
HERE	002674	1731#													
ILFT	000544	1357#	2256												
INIT	017002	2940	2961	2995	3037	3083	3128	3603#							
INIT1	016724	1834	1939	1981	2007	2058	2069	2099	2108	2131	2160	2257	2284	2314	
		2338	3580#												
INIT2	016734	2183	2205	2231	3582#										
INIT3	016752	2364	2420	2445	2491	2512	2522	2568	2592	2714	2751	2769	2791	2802	
		2817	2823	2838	2845	2888	3168	3201	3592#						
INIT4	016766	2777	2865	3233	3597#										
INIT5	017006	3594	3599	3604#											
INMT	016646	2156	2179	2201	2227	3558#									
INST	017176	1752	1874	2159	2182	2204	2230	3637#							
ITAMT	000602	1375#	2363*	2413*	2567*	2613*	3550								
ITCNT	000676	1408#	3349	3513	3545*	3549*	3550								
ITER	016576	1825	1859	1971	1997	2023	2050	2091	2123	2142	2168	2191	2213	2239	
		2275	2306	2329	2354	2412	2437	2478	2502	2555	2611	2737	2760	2783	
		2809	2830	2857	2880	2922	2983	3027	3072	3117	3160	3192	3222	3254	
		3543#													
ITRLP	000710	1413#	1700*	1701*	3553										
JUMPER	001016	1448#	1626	1631											
LTADD	000736	1424#	1698*	1699*	1700	1703	1719*	1720*							
LTGA	015244	3311#	3357												
LTGA1	015262	3312	3315#												
LTGA2	015314	3319	3322#												
LTGB	015342	3326	3328#												
LTGC	015346	3323	3329#												
LTGCO	015434	3333	3344#												
LTGD	015440	3331	3345#												
LTGER	015226	1785	1821	1931	2497	3307#									
LTGERO	015220	2266	2295	2324	2349	2400	2432	2472	2521	2547	2732	2759	2776	2782	
		2801	2808	2822	2829	2844	2856	2879	2900	2905	2910	2976	3013	3057	
		3103	3149	3217	3249	3306#									
LTGER1	016334	1848	1852	1965	1969	1995	2020	2048	2078	2084	2090	2167	2190	2212	
		2238	3021	3065	3110	3191	3488#								
LTGER2	016326	2114	2118	2122	2141	3487#									
LTGER3	015206	2270	2299	2328	2353	2408	2436	2477	2554	2585	2608	2736	2916	2957	

		2982	3026	3071	3116	3159	3221	3253	3304#
LTGX	015470	3310	3346	3351#					
LTGXA	015500	3352	3354#						
LTGXX	015516	3355	3358#						
LTG1A	016352	3492#	3523						
LTG1B	016370	3493	3496#						
LTG1C	016500	3510	3515#						
LTG1T	016450	3500	3509#						
LTG1X	016504	3491	3516#						
LTG1XX	016534	3521	3524#						
LTG1X1	016516	3518	3520#						
LT1	002734	1476	1477	1742#					
LT1A	003060	1750	1764#	1779	1784				
LT1B	003126	1768	1774#						
LT1C	003134	1772	1776#	1786					
LT1ER	003144	1773	1780#						
LT1ER1	003154	1775	1782#						
LT1ER2	003162	1781	1783#						
LT1G	003010	1753#	1777						
LT1G0	003000	1751#							
LT1X	003204	1748	1761	1770	1787#				
LT10	004620	1490	2029#						
LT10A	004636	2033#	2045	2047					
LT10A1	004634	2032#	2043						
LT10B	004672	2040#	2049						
LT10E1	004712	2039	2046#						
LT10IT	004622	1491	2030#						
LT10X	004734	2041	2050#						
LT11	004744	1492	2056#						
LT11B	004772	2062#	2079	2083					
LT11C	005032	2069#	2085	2089					
LT11E1	005052	2061	2074#						
LT11E2	005102	2068	2080#						
LT11E3	005132	2073	2086#						
LT11IT	004746	1493	2057#	2077					
LT11X	005160	2072	2091#						
LT12	005170	1494	2097#						
LT12B	005214	2102#	2115	2117					
LT12C	005252	2108#	2119	2121					
LT12E1	005270	2101	2112#						
LT12E2	005312	2107	2116#						
LT12E3	005334	2111	2120#						
LT12IT	005172	1495	2098#	2113					
LT12X	005354	2110	2123#						
LT13	005364	1496	2129#						
LT13A	005424	2136#	2137						
LT13E1	005430	2138#							
LT13IT	005374	1497	2131#	2140					
LT13X	005456	2132	2142#						
LT14	005466	1498	2152#						
LT14A	005510	2153	2157#						
LT14IT	005526	1499	2160#	2165					
LT14X	005566	2164	2168#						
LT14XX	005572	2155	2169#						
LT15	005576	1500	2175#						
LT15A	005620	2176	2180#						

LT15IT	005636	1501	2183#	2188
LT15X	005676	2187	2191#	
LT15XX	005702	2178	2192#	
LT16	005706	1502	2197#	
LT16A	005730	2198	2202#	
LT16IT	005746	1503	2205#	2210
LT16X	006006	2209	2213#	
LT16XX	006012	2200	2214#	
LT17	006020	1504	2223#	
LT17A	006042	2224	2228#	
LT17IT	006060	1505	2231#	2236
LT17X	006120	2235	2239#	
LT17XX	006124	2226	2240#	
LT2	003210	1478	1479	1792#
LT2A	003250	1799#	1809	1824
LT2B	003270	1802	1804#	
LT2C	003304	1805	1807#	
LT2ERG	003364	1814	1817	1820#
LT2ER1	003314	1803	1812#	
LT2ER2	003332	1806	1815#	
LT2ER3	003350	1818#		
LT2IT	003212	1793#		
LT2LP	003400	1820	1823#	
LT2X	003404	1810	1825#	
LT20	006132	1506	2253#	
LT20A	006160	2254	2257#	2274
LT20B	006236	2263	2268#	
LT20C	006246	2267	2269	2271#
LT20IT	006146	1507	2255#	
LT20X	006260	2272	2275#	
LT21	006274	1508	2282#	
LT21A	006400	2292	2297#	
LT21B	006410	2296	2298	2301#
LT21IT	006310	1509	2283	2284#
LT21XA	006414	2302#	2303	
LT22	006440	1510	2312#	
LT22A	006540	2321	2326#	
LT22IT	006454	1511	2313	2314#
LT22X	006550	2325	2327	2329#
LT23	006564	1512	2336#	
LT23A	006664	2346	2351#	
LT23IT	006600	1513	2337	2338#
LT23X	006674	2350	2352	2354#
LT24	006714	1514	2361#	
LT24B	007042	2380#	2386	
LT24B0	007064	2383	2385#	
LT24C	007076	2389#	2390	
LT24D	007160	2392	2403#	
LT24IT	006730	1515	2362	2363#
LT24X	007204	2402	2407	2409#
LT25	007240	1516	2419#	
LT25A	007346	2428	2434#	
LT25IT	007246	1517	2420#	2429
LT25X	007356	2433	2435	2437#
LT26	007372	1518	2444#	
LT26IT	007400	1519	2445#	2469

LT26X	007572	2473	2476	2478#		
LT27	007606	1520	2485#			
LT27A	007642	2487	2491#	2501		
LT27B	007700	2494	2498#			
LT27IT	007632	1521	2489#			
LT27X	007710	2499	2502#			
LT27XX	007720	2486	2504#			
LT3	003414	1480	1830#			
LT3A	003434	1834#	1844	1846	1858	
LT3B	003454	1838#	1849	1851		
LT3C	003470	1841#	1853			
LT3ER1	003500	1837	1845#			
LT3ER2	003526	1840	1850#			
LT3IT	003416	1481	1831#			
LT3X	003550	1842	1854#			
LT3XX	003566	1855	1859#			
LT30	007724	1522	2509#			
LT30A	010026	2519	2522#			
LT30B	010070	2529#	2532			
LT30C	010130	2530	2539#			
LT30D	010146	2542#	2543			
LT30E	010176	2545	2549#			
LT30IT	007746	1523	2511	2512#		
LT30X	010222	2538	2548	2553	2555#	
LT31	010242	1524	2564#			
LT31A	010420	2590	2592#			
LT31IT	010250	1525	2565#			
LT31X	010524	2586	2588	2609	2611#	2659
LT32	011072	1526	2712#			
LT32IT	011106	1527	2713	2714#		
LT32X	011234	2733	2735	2737#		
LT32XX	011244	2725	2739#			
LT33	011250	1528	2749#			
LT33IT	011264	1529	2750	2751#		
LT33X	011342	2756	2760#			
LT34	011352	1530	2766#			
LT34A	011412	2772#				
LT34A1	011372	2769#	2775			
LT34B	011442	2773	2777#			
LT34C	011446	2778#	2780			
LT34IT	011366	1531	2768#			
LT34X	011476	2779	2783#			
LT34XX	011502	2784#				
LT35	011506	1532	2789#			
LT35A	011610	2798	2802#	2806		
LT35IT	011522	1533	2791#	2800		
LT35X	011650	2805	2809#			
LT36	011660	1534	2815#			
LT36IT	011674	1535	2817#	2820		
LT36X	011770	2826	2830#			
LT37	012000	1536	2836#			
LT37A	012054	2841	2845#	2854		
LT37B	012106	2850#	2853			
LT37IT	012014	1537	2838#	2843		
LT37X	012140	2851	2857#			
LT4	003576	1482	1483	1865#		

LT4A	003714	1872	1886#	1924	1930
LT4B	003756	1891	1895#		
LT4C	003764	1897#			
LT4D	004116	1893	1921#	1932	
LT4ERG	004144	1926	1928#		
LT4ER1	004126	1894	1925#		
LT4ER2	004136	1896	1927#		
LT4G	003644	1875#	1922		
LT4G0	003634	1873#			
LT4X	004174	1870	1883	1920	1933#
LT40	012150	1538	2863#		
LT40IT	012164	1539	2863	2865#	
LT40X	012260	2874	2880#		
LT40XX	012264	2881#			
LT41	012270	1540	2886#		
LT41IT	012304	1541	2886	2888#	
LT41X	012500	2911	2915	2917#	
LT42	012532	1542	2933#		
LT42A	012634	2947#	2950		
LT42B	012650	2948	2951#		
LT42B1	012676	2952	2957#		
LT42B2	012716	2956	2961#		
LT42C	012762	2968#	2971		
LT42D	012776	2969	2972#		
LT42E	013026	2973	2978#		
LT42IT	012570	1543	2939	2940#	
LT42X	013046	2977	2981	2983#	
LT43	013062	1544	2989#		
LT43C	013172	3005#	3008		
LT43D	013206	3006	3009#		
LT43E	013240	3010	3015#		
LT43F	013272	3019	3022#		
LT43IT	013120	1545	2993	2995#	
LT43X	013312	3014	3025	3027#	
LT43XX	013322	2990	3029#		
LT44	013326	1546	3033#		
LT44A	013412	3043#	3046		
LT44A1	013424	3044	3047#		
LT44B	013434	3049#	3052		
LT44C	013450	3050	3053#		
LT44D	013476	3054	3058#		
LT44E	013500	3059#			
LT44F	013540	3060	3067#		
LT44IT	013354	1547	3036	3037#	
LT44X	013560	3066	3070	3072#	
LT44XX	013570	3074#			
LT45	013574	1548	3079#		
LT45A	013702	3095#	3098		
LT45B	013716	3096	3099#		
LT45D	013746	3100	3104#		
LT45E	013660	3089#	3092		
LT45E1	013672	3090	3093#		
LT45F	014000	3108	3111#		
LT45IT	013622	1549	3082	3083#	
LT45X	014024	3115	3117#		
LT45XX	014034	3119#			

CZTEADO TMO3-TE16/TU77 CTL I
CZTEAD.P11 06-JUL-83 14:40

MACY11 30(1046) 06-JUL-83 21:01 K 10
PAGE 87-7
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0127

MSG10	021564	1850	3949#			
MSG11	021610	1845	3950#			
MSG12	021635	3336	3501	3951#		
MSG13	021644	3340	3505	3952#		
MSG14	021653	1963	3953#			
MSG15	021670	1967	3954#			
MSG16	021706	3347	3511	3955#		
MSG18	021716	1993	3956#			
MSG19	021735	2018	3189	3957#		
MSG2	021045	1753	3937#			
MSG2A	021106	1751	3938#			
MSG20	021754	2046	3958#			
MSG21	022006	2074	3959#			
MSG22	022044	2080	3960#			
MSG23	022065	2086	3961#			
MSG24	022120	2112	3962#			
MSG25	022146	2116	3963#			
MSG25A	022175	2120	3964#			
MSG26	022222	2139	3965#			
MSG27	022251	2166	2189	2211	2237	3966#
MSG3	021235	1783	3940#			
MSG30	022265	1915	3967#			
MSG31	022273	3452	3968#			
MSG32	022311	3448	3969#			
MSG33	022327	3455	3970#			
MSG34	022350	3971#				
MSG35	022365	3972#				
MSG36	022403	3973#				
MSG37	022421	3974#				
MSG4	021257	1813	1819	3941#		
MSG40	022437	3757	3975#			
MSG41	022443	1724	3976#			
MSG42	022462	3064	3977#			
MSG43	022477	3559	3978#			
MSG44	022603	1591	3980#			
MSG45	022625	1600	3981#			
MSG46	022647	3365	3982#			
MSG47	022753	3428	3984#			
MSG5	021275	1816	3942#			
MSG50	023005	2535	3985#			
MSG51	023024	3305	3986#			
MSG53	023053	3020	3987#			
MSG54	023065	3109	3988#			
MSG55	023076	2958	3989#			
MSG56	023173	1652	3990#			
MSG57	023220	1634	3991#			
MSG58	023377	1643	3994#			
MSG59	023422	3716	3995#			
MSG6	021315	3324	3943#			
MSG60	023424	3725	3996#			
MSG62	023426	1582	3997#			
MSG63	023474	3672	3998#			
MSG64	023522	1897	3999#			
MSG65	023540	1900	4000#			
MSG66	023545	1903	4001#			
MSG67	023552	1662	4002#			

CZTEAD0 TMO3-TE16/TU77 CTL I
CZTEAD.P11 06-JUL-83 14:40

MACY11 30(1046) 06-JUL-83 21:01 PAGE 88
D 11
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0133

\$CATCH	1226#	1308
\$CHAIN	1226#	1578
\$CHNMO	1226#	1697
\$RESTO	1226#	3930
\$SAVE	1226#	3929
.\$ACT1	1226#	1309
.\$EOP	1226#	1731

. ABS. 031370 000

ERRORS DETECTED: 0

CZTEAD,CZTEAD/CRF=CZTEAD.SML/ML,CZTEAD.P11
RUN-TIME: 5 9 1 SECONDS
RUN-TIME RATIO: 24/16=1.4
CORE USED: 14K (28 PAGES)