.REM *

IDENTIFICATION

PRODUCT CODE: AC A791E-MC

PRODUCT NAME: CZTEAE0 TM03-TE16/TU77 CONTROL LOGIC TEST PART I

DATE CREATED: 15 MARCH 1981

MAINTAINER: TAPE DIAGNOSTIC ENGINEERING

AUTHOR: J. HITT

TABLE OF CONTENTS

1. ABSTRACT

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

2. REQUIREMENTS (HARDWARE)

A. ANY PDP-11 PROCESSOR
B. 8K OF CORE
C. CONSOLE TTY
D. TM03 MAGTAPE CONTROLLER
E. MASSBUS CONTROLLER (RH)
F. 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 IMDP 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 RESONSES ARE SHOWN IN PARENTHESES (), AND
MEMORY LOCATIONS CONTAINING THE DEFAULT ARE SHOWN IN
SQUARE BRACKETS [].
IN THIS EXAMPLE THE OPERATOR HAS CHOSEN DEFAULT RESPONSES,
TO INVOKE THE DEFAULT TYPE (CR).

** NON-STANDARD JUMPER MODE
M8931 (W2 IN) ,M8932 (W2 IN,W1 OUT) **

```
PARAMETER REQUEST: <DEFAULT> (RESONSE)              [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 <S>      [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)                [SLV.TYP:]
        IF THE SOFTWARE SWR IS INVOKED:
SWR = <000000> NEW = (CR)                           [SWREG:]
```

LOGIC TEST #23: FORMAT ERROR (FMT)

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

LIKELY FAULT LOCATIONS: M8905-YB,M8906,M8909

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

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

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

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

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

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

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

LIKELY FAULT LOCATION: M8909

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

LOGIC TEST #26: FRAME COUNT ERROR

PROGRAMMED SEQUENCE: SET WC TO -10, FC TO -20 WAM3 IN MAINTENANCE REGISTER, LOAD WRITE AND GO, DELAY ISSUE MM OR CLEAR. CHECK FCE AND CHECK FOR UNEXPECTED ERRORS. FRAME COUNT ERROR SHOULD BE SET BECAUSE A WRITE OPERATION WAS TERMINATED PRIOR TO A WORD COUNT OVERFLOW.

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

CIRCUITS	PRINT REFERENCE
RUN LINE	MB1
EBL PLS	MBI9
FCE FLOP	MBI11
SHUTDOWN LOGIC	TCCM5
MAINT. FUNCTION DECODE	MR5

LOGIC TEST #27: ILLEGAL REGISTER

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

LIKELY FAULT LOCATIONS: MASSBUSS, M8909

CIRCUITS	PRINT REFERENCE
REGISTER SELECT LINES	MB1, MB2
REGISTER SELECT DECODE	MBI2
ILR FLOP	MBI11

LOGIC TEST #30: DRIVE TIMING ERROR

PROGRAMMED SEQUENCE:

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

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

CIRCUITS	PRINT REFERENCES
DTE FLOP	MBI11
CRIPPLE OCCUPIED FUNCTION	MR5
WRP 3 FUNCTION	MR5
PREVIOUS OCCUPIED CHECK	MBI7
CHECK FOR WCLK	BF.
MM CLK	MR5

LOGIC TEST 31: OPERATION INCOMPLETE (OPI)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8933, M8909

CIRCUITS	PRINT REFERENCES
OPI TIMER	ICCM5
OPI FLOP	MBI11
OPI TIMER CONTROL	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	MBI11
SET UNSAFE	MBI7
MUL GENERATION	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	M8I5
PIP GENERATION	TCCM7
STATUS REGISTER	TCCM7

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

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

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

CIRCUITS	PRINT REFERENCES
DENSITY BITS	MR6
DENSITY LINES	SBC
PES CIRCUIT	SC3
PES STATUS BIT	TCCM7

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)(ICCM7)(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	(M8937,2,3),(LAW6)
TE16 ADDRESS DECODE	(LAW6)
SPR BIT	(LAW6)(M9001YA)(ICCM7)

```
LT4A    003716  1875  1889#  1927  1933
LT4B    003760  1894  1898#
LT4C    003766  1900#
LT4D    004120  1896  1924#  1935
LT4ERG  004146  1929  1931#
LT4ER1  004130  1897  1928#
LT4ER2  004140  1899  1930#
LT4G    003646  1878#  1925
LT4G0   003636  1876#
LT4X    004176  1873  1886  1923  1936#
LT40    012152  1540  2866#
LT40IT  012166  1541  2866  2868#
LT40X   012262  2877  2883#
LT40XX  012266  2884#
LT41    012272  1542  2889#
LT41IT  012306  1543  2889  2891#
LT41X   012502  2914  2918  2920#
LT42    012534  1544  2936#
LT42A   012636  2950#  2953
LT42B   012652  2951  2954#
LT42B1  012700  2955  2960#
LT42B2  012720  2959  2964#
LT42C   012764  2971#  2974
LT42D   013000  2972  2975#
LT42E   013030  2976  2981#
LT42IT  012572  1545  2942  2943#
LT42X   013050  2980  2984  2986#
LT43    013064  1546  2992#
LT43C   013174  3008#  3011
LT43D   013210  3009  3012#
LT43E   013242  3013  3018#
LT43F   013274  3022  3025#
LT43IT  013122  1547  2996  2998#
LT43X   013314  3017  3028  3030#
LT43XX  013324  2993  3032#
LT44    013330  1548  3036#
LT44A   013414  3046#  3049
LT44A1  013426  3047  3050#
LT44B   013436  3052#  3055
LT44C   013452  3053  3056#
LT44D   013500  3057  3061#
LT44E   013502  3062#
LT44F   013542  3063  3070#
LT44IT  013356  1549  3039  3040#
LT44X   013562  3069  3073  3075#
LT44XX  013572  3077#
LT45    013576  1550  3082#
LT45A   013704  3098#  3101
LT45B   013720  3099  3102#
LT45D   013750  3103  3107#
LT45E   013662  3092#  3095
LT45F1  013674  3093  3096#
LT45F   014002  3111  3114#
LT45IT  013624  1551  3085  3086#
LT45X   014026  3118  3120#
LT45XX  014036  3122#
```

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

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

LOGIC TEST #13: INTERRUPT TEST

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

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

LIKELY FAULT LOCATION:

CIRCUITS	PRINT REFERENCE
INTERRUPT CONTROL	BCTF

MANUAL INTERVENTION TESTS 14,15,16,17

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

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

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

LIKELY FAULT LOCATION: M8910,SLAVE CABLE, M8933

CIRCUIT	PRINT REFERENCE
MOL	LAW6,TCCM7,M8908,M9001-A,-C
WRL	LAW8,TCCM7,M8908,M9001-A,-C
DPR	TCCM7
DRY	TCCM7
BOT	LAW6,TCCM7,M8908-A,M8913,-A

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	LAW8,M8913,M8913YA,TCCM7
ATA	M8I3

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	LAW8,M8913,M8913YA,TCCM7
EOT	LAW6,TCCM7,M8908YA,M8913YA
SLA	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	MBI5,MBI7
ILF FLOP	MBI11
ILF MULTIPLEXER	MBI10

LOGIC TEST #21: REGISTER MODIFICATION REFUSED

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

LIKELY FAULT LOCATION: M8909

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

LOGIC TEST #22: CONTROL BUS PARITY (CPAR)

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

LIKELY FAULT LOCATIONS: M8909

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

LOGIC TEST #23: FORMAT ERROR (FMT)

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

LIKELY FAULT LOCATIONS: M8905-YB,M8906,M8909

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

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

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

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

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

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

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

LIKELY FAULT LOCATION: M8909

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

LOGIC TEST #26: FRAME COUNT ERROR

PROGRAMMED SEQUENCE: SET WC TO -10, FC TO -20 WAM3 IN MAINTENANCE REGISTER, LOAD WRITE AND GO, DELAY ISSUE MM OR CLEAR. CHECK FCE AND CHECK FOR UNEXPECTED ERRORS. FRAME COUNT ERROR SHOULD BE SET BECAUSE A WRITE OPERATION WAS TERMINATED PRIOR TO A WORD COUNT OVERFLOW.

LIKELY FAULT LOCATIONS: M8909, M8 CABLE, M8933, M8905-Y8

CIRCUITS	PRINT REFERENCE
RUN LINE	MB1
EBL PLS	MBI9
FCE FLOP	MBI11
SHUTDOWN LOGIC	TCCM5
MAINT. FUNCTION DECODE	MR5

LOGIC TEST #27: ILLEGAL REGISTER

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

LIKELY FAULT LOCATIONS: MASSBUSS, M8909

CIRCUITS	PRINT REFERENCE
REGISTER SELECT LINES	MB1, MB2
REGISTER SELECT DECODE	MBI2
ILR FLOP	MBI11

LOGIC TEST #30: DRIVE TIMING ERROR

PROGRAMMED SEQUENCE:

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

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

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

LOGIC TEST 31: OPERATION INCOMPLETE (OPI)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8933, M8909

CIRCUITS	PRINT REFERENCES
OPI TIMER	TCCM5
OPI FLOP	MBI11
OPI TIMER CONTROL	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	MBI11
SET UNSAFE	MBI7
MUL GENERATION	LAW6

SEQ 0022

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	M8I5
PIP GENERATION	TCCM7
STATUS REGISTER	TCCM7

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

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

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

CIRCUITS	PRINT REFERENCES
DENSITY BITS	MR6
DENSITY LINES	SBC
PES CIRCUIT	SC3
PES STATUS BIT	TCCM7

LOGIC TEST 35: TAPE CONTROL WRITE (TCW)

PROGRAMMED SEQUENCE:

SETUP FORMAT AND WRP-3 ARE SET, READ COMMAND IS ISSUED. TCW IS CHECKED. DRIVE IS INITIALIZED, TAPE CONTROL REGISTER 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	M8108
FCS MULTIPLEXER	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	TCCM3
CLOCK	SC2

LOGIC TEST 40: PE TAPE MARK (TM)

PROGRAMMED SEQUENCE:

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

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

CIRCUITS	PRINT REFERENCES
TAPE MARK DETECTOR	TCPE4, TCPE5
TAPE MARK MULTIPLEXER	TCCM7
ENVELOPE SIGNALS	DS 3, 5, 7
WRITE DATA BUFFER	TCCM5
RDA MULTIPLEXERS	TCCM6
WRITE TAPE MARK FUNCTION	MB15
WAMO SIGNAL	MR5

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

PROGRAMMED SEQUENCE:

SAME AS TEST 40 EXCEPT NRZ DENSITY IS SELECTED.

LIKELY FAULT LOCATIONS: M8933, M8934

CIRCUITS	PRINT REFERENCES
WRITE DATA BUFFER	TCCM2
RSDO MULTIPLEXER	TCCM6
RDA MULTIPLEXERS	TCCM6
TM DETECTOR	CNRZ4
ILLEGAL TAPE MARK FLOP	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 WRAPO DESIGNED TO LOAD THE CRC CHECKER IN A KNOWN MANNER. CHECK ARE MADE FOR LRC ERROR AND THE CONTENT OF CRC REGISTER. THEN A WRITE OPERATION IS PERFORMED USING A MAINT. MODE (IICC) WHICH INHIBITS THE INITIALIZATION OF THE CRC CHECKER. THE CRC CHECKER LOGIC WHICH HAS NOT BEEN CLEARED SHOULD DETECT A CRC ERROR. UNEXPECTED ERROR BITS MAY INDICATE PROBLEMS WITH THE WRITE OPERATION.

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

CIRCUITS	PRINT REFERENCES
MM FUNCTION DECODE	MR5
CRC CHECK CIRCUIT	CNRZ3

LOGIC TEST 43: LRC

PROGRAMMED SEQUENCE:

A WRITE OPERATION IS PERFORMED WITH A MM FUNCTION (INC IMRL) WHICH ASSERTS WD(SB) 5L THROUGHOUT THE RECORD. ALL ONES DATA IS USED SO THAT THE FUNCTION ONLY INTERFERES WITH THE WRITING OF THE LRC CHARACTER WHEN NONE OF THE TM03 WRITE DATA LINES SHOULD BE ASSERTED.
** NOTE: THIS TEST IS NOT PERFORMED ON A TU77 SLAVE.

LIKELY FAULT LOCATIONS: M8505, M8933, M8910, M8934

CIRCUITS	PRINT REFERENCES
MM FUNCTION DECODE	MR5
WRITE LINE DRIVERS	[illegible]
WRITE HEAD DRIVERS	[illegible], 4
LRC CHECKING	CNRZ3

LOGIC TEST 44: PE CORRECTABLE DATA

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905-YB, M8901, M8932

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

LOGIC TEST 45: PE INCORRECTABLE DATA

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

LIKELY FAULT LOCATIONS: M8932, M8901

CIRCUIT	PRINT REFERENCE
INC ERROR, PEF,	TCPE2

LOGIC TEST 46: PE FORMAT

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

LIKELY FAULT LOCATIONS: M8932, M8933, M8905-YB

CIRCUITS	PRINT REFERENCES
PEF.	TCPE2
WRITE BUFFER	TCCM2
MM DECODE	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 M8I8

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.

SEQ 0029

```
                        .LIST   BIN,LOC,SEQ
                        .TITLE C2TEAE0 TM03-TE16/TU77 CTL I
                        ;CONTROL LOGIC TEST PART I
                        ;AC-A791E-MC
                        ;FEB 77
                        ;J.G. ADAMS
                        ;REVISED MAY 1978 BY J. G. ADAMS ;++B CHANGED MODULE REFERENCES TO
                                                         ;++B REFLECT TM03 MODULES
                                                        ;++B ADDED TU77 TEST CAPABILITY
                        ;REVISED NOV 1978 BY M. PAGE     ;+ INDICATES ENHANCEMENTS TO
                        ;                                  THE ORIGINAL REV (DZTEAA)
                        ;                                  NECESSARY FOR TM03
                        ;REVISED MAY,1983 BY B. LEBLANC ;BL FIXED AIDS #CC0001220
                        ;REVISED MARCH 15,1984 BY J. HITT       ;ADD XON/XOFF FUNCTIONALITY
                                                                ;MAKE TEXT CHANGES

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


                        ;CONSOLE SWITCHES*********
                        ;
                        ;SW15:  1=HALT ON ERROR
                        ;       0=CONTINUE
                        ;SW14:  1=LOOP ON ERROR
                        ;       0=CONTINUE
                        ;SW13:  1=DO NOT PRINT ERRORS
                        ;       0=PRINT ERRORS
                        ;SW12:  0=CONTINUOUS CYCLE
                        ;       1=HALT AT END OF PASS
                        ;SW11:  1=INHIBIT ITERATIONS
                        ;       0=DO ITERATIONS
                        ;SW10:  1=HALT AT END OF EACH TEST
                        ;       0=CONTINUE
                        ;SW9:   1=DO MANUAL INTERVENTION TESTS
                        ;       0=INHIBIT MANUAL INTERVENTION
                        ;SW0-5: SELECT TEST NUMBER :: 00=ALL TESTS
```

```
                        ;TE16 REGISTER BITS**********

                        ;15 ;14 ;13 ;12 ;11 ;10 ;09 ;08 ;07 ;06 ;05 ;04 ;03 ;02 ;01 ;00 ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;CS1:   ;SC ;TRE;MCP;SPR;DNA;PSL;A17;A16;RDY;IE ;FUN;FUN;FUN;FUN;GO ;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;WC:    ; WORD                                                          ;
                        ; COUNT                                                         ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;BA:    ; BUS                                                           ;
                        ; ADDRESS                                                       ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;FC:    ; FRAME                                                         ;
                        ; COUNT                                                         ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;CS2:   ;DLT;WCE;UPE;NED;NEM;PGE;MXF;MDP;OR ;IR ;CLR;PAT;BAI;U2 ;U1 ;U0 ;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;ERR;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;DS:    ;ATA;ERR;PIP;MOL;WRL;EOT;XXX;DPR;DRY;SSC;PES;SDN;IDB;TMK;BOT;SLA;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;ER:    ;CDE;UNS;OPI;DTE;NEF;CS ;FCE;NSG;DEF;INC;DAT;FMT;CNT;RMR;ILR;ILF;
                        ;CRC;   ;   ;   ;ITM;   ;   ;LRC;VPE;BPE;   ;BPE;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;ATS:   ; ATTENTION                                                     ;
                        ; SUMMARY                                                       ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;CC:    ; CHECK                                                         ;
                        ; CHARACTER                                                     ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;DB:    ; DATA                                                          ;
                        ; BUFFER                                                        ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;MR:    ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;WRT;MM ;OP ;OP ;OP ;OP ;MM ;
                        ; 7 ; 6 ; 5 ; 4 ; 3 ; 2 ; 1 ; 0 ; P ;CLK;CLK; 4 ; 3 ; 2 ; 1 ;GO ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;DT:    ; DRIVE      ; 7 ;    ;SLV;                                   ;
                        ; TYPE       ;CH ;    ;PR ;                                   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;SN:    ; SERIAL                                                        ;
                        ; NUMBER                                                        ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;TC:    ;ACL;FCS;SAC;ENA;SPR;DEN;DEN;DEN;FMT;FMT;FMT;FMT;EVN;SSN;SSN;SSN;
                        ;   ;   ;   ;DT ;   ; 4 ; 2 ; 1 ; 8 ; 4 ; 2 ; 1 ;PAR; 4 ; 2 ; 1 ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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

```
                        000510                          .=510
                                                        ;MASS BUS REGISTER EQUIVS**********
                                                
000510  172440                        C1:     172440
000512  172442                        WC:     172442
000514  172444                        BA:     172444
000516  172446                        FC:     172446
000520  172450                        CS:     172450
000522  172452                        DS:     172452
000524  172454                        ER:     172454
000526  172456                        AS:     172456
000530  172460                        CC:     172460
000532  172462                        DB:     172462
000534  172464                        MR:     172464
000536  172466                        DT:     172466
000540  172470                        SN:     172470
000542  172472                        TC:     172472
                                                
                                                        ;ILLEGAL FUNCTION CODES
                                                
000544  005405                        ILFT:   5405
000546  007415                                7415
000550  016423                                16423
000552  020437                                20437
000554  022443                                22443
000556  025447                                25447
000560  031455                                31455
000562  033465                                33465
000564  036473                                36473
                                                
                                                        ;CONSTANTS**********
                                                
000566  177776                PSW:    177776          ;PROCESSOR STATUS
000570  177570                SWR:    177570          ;SWITCH REGISTER
000572  177560                TKS:    177560          ;TTY READER STATUS
000574  177562                TKB:    177562          ;TTY READ BUFFER
000576  177564                TPS:    177564          ;TTY PUNCH STATUS
000600  177566                TPB:    177566          ;TTY PUNCH BUFFER
000602  000020                ITAMT:  20              ;ITERATION AMOUNT
000604  000224                VECT:   224             ;INTERRUPT VECTOR(RH)
000606  172440                REGS:   172440          ;STARTING REGISTER ADDRESS
```

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

```
1437  000764  000000                PREFL:  0
1438  000766  000000                SERFL:  0
1439  000770  000000                CRCNT:  0
1440  000772  000000                UDES:   0
1441  000774  000000                WPGFL:  0
1442  000776  000000                PATRN:  0
1443  001000  000000                STATF:  0
1444  001002  000000                RDRVF:  0
1445  001004  000000                RCDP:   0
1446  001006  000000                STATC:  0
1447  001010  000000                SLVTYP: .WORD   0               ;**B INDICATES SLAVE TYPE (0/1 = TE16/TU77)
1448  001012  000000                SKAT:   0
1449  001014  000000                PCNTR:  0               ;PASS COUNTER
1450  001016  000003                JUMPER: 3               ;*INDICATOR FOR NON-STANDARD CONFIG.
1451  001020  000000                NONSTD: 0               ;*FLAG FOR NON-STANDARD CONFIG.
1452
1453                                      ;EXPT WRAP STATUS*******
1454
1455  001022  000000                WCS1:   0
1456  001024  000000                WCS2:   0
1457  001026  000000                WDS:    0
1458  001030  000000                WER:    0
1459
1460                                      ;CORE DUMP PATTERNS*******
1461
1462  001032  000005                WCDP2:  5
1463  001034  000005                        5
1464  001036  000012                        12
1465  001040  000012                        12
1466  001042  000000                        0
1467  001044  000017                WCDP0:  17
1468  001046  000017                        17
1469  001050  000017                        17
1470  001052  000017                        17
1471  001054  000000                        0
```

```
1473
1474                                    ;LOGIC TEST ENTRY TABLE**********
1475
1476  001056  000000            TSTTBL: 0
1477  001060  000000                    0
1478  001062  002736                    LT1
1479  001064  002736                    LT1
1480  001066  003212                    LT2
1481  001070  003212                    LT2
1482  001072  003416                    LT3
1483  001074  003420                    LT3IT
1484  001076  003600                    LT4
1485  001100  003600                    LT4
1486  001102  004202                    LT5
1487  001104  004210                    LT5IT
1488  001106  004372                    LT6
1489  001110  004374                    LT6IT
1490  001112  004506                    LT7
1491  001114  004510                    LT7IT
1492  001116  004622                    LT10
1493  001120  004624                    LT10IT
1494  001122  004746                    LT11
1495  001124  004750                    LT11IT
1496  001126  005172                    LT12
1497  001130  005174                    LT12IT
1498  001132  005366                    LT13
1499  001134  005376                    LT13IT
1500  001136  005470                    LT14
1501  001140  005530                    LT14IT
1502  001142  005600                    LT15
1503  001144  005640                    LT15IT
1504  001146  005710                    LT16
1505  001150  005750                    LT16IT
1506  001152  006022                    LT17
1507  001154  006062                    LT17IT
1508  001156  006134                    LT20
1509  001160  006150                    LT20IT
1510  001162  006276                    LT21
1511  001164  006312                    LT21IT
1512  001166  006442                    LT22
1513  001170  006456                    LT22IT
1514  001172  006566                    LT23
1515  001174  006602                    LT23IT
1516  001176  006716                    LT24
1517  001200  006732                    LT24IT
1518  001202  007242                    LT25
1519  001204  007250                    LT25IT
1520  001206  007374                    LT26
1521  001210  007402                    LT26IT
1522  001212  007610                    LT27
1523  001214  007634                    LT27IT
1524  001216  007726                    LT30
1525  001220  007750                    LT30IT
1526  001222  010244                    LT31
1527  001224  010252                    LT31IT
1528  001226  011074                    LT32
```

```
1529  001230  011110                  LT32II
1530  001232  011252                  LT33
1531  001234  011266                  LT33II
1532  001236  011354                  LT34
1533  001240  011370                  LT34II
1534  001242  011510                  LT35
1535  001244  011524                  LT35II
1536  001246  011662                  LT36
1537  001250  011676                  LT36II
1538  001252  012002                  LT37
1539  001254  012016                  LT37II
1540  001256  012152                  LT40
1541  001260  012166                  LT40II
1542  001262  012272                  LT41
1543  001264  012306                  LT41II
1544  001266  012534                  LT42
1545  001270  012572                  LT42II
1546  001272  013064                  LT43
1547  001274  013122                  LT43II
1548  001276  013330                  LT44
1549  001300  013356                  LT44II
1550  001302  013576                  LT45
1551  001304  013624                  LT45II
1552  001306  014042                  LT46
1553  001310  014070                  LT46II
1554  001312  014304                  LT47
1555  001314  014320                  LT47II
1556  001316  014474                  LT50
1557  001320  014510                  LT50II
1558  001322  014650                  LT51
1559  001324  014664                  LT51II
1560  001326  002622  TADX:    .WORD   TEND
1561  001330  000051  TLAST:   .WORD   51          ;CONTAINS # OF TESTS
1562  001332  000000  $CNTRLS: .WORD   0           ;XON/XOFF FLAG
```

```
                                        .EVEN
                                        ;PROGRAM START AND HOUSEKEEPING*********

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

M3


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

N3


```
1668  002154  004737  020172                  JSR     PC,OCTP          ;++B TYPE CURRENT VALUE
1669  002160  012705  001010                  MOV     #SLVTYP,R5       ;++B GET ADDRESS OF SLVTYP FLAG
1670  002164  012701  000002                  MOV     #2,R1            ;++B SET SIZE OF RESPONSE
1671  002170  012702  000001                  MOV     #1,R2            ;++B SET UPPER LIMIT
1672  002174  012703  000000                  MOV     #0,R3            ;++B SET LOWER LIMIT
1673  002200  004737  017436                  JSR     PC,TTR           ;++B GET RESPONSE
1674  002204  005737  001010                  TST     SLVTYP           ;IS IT A TU77
1675  002210  001406                          BEQ     ST2              ;BRANCH IF NOT
1676  002212  012737  116741  006020          MOV     #116741,STWD16   ;SET UP TEST WORD FOR TEST 16
1677  002220  012737  110741  006132          MOV     #110741,STWD17   ;SET UP TEST WORD FOR TEST 17
1678
1679
1680                          ;START 210
1681  002226  012706  000500          ST2:    MOV     #500,SP          ;SET STACK PTR
1682  002232  005037  001302                  CLR     RDRVF            ;CLEAR REVERSE FLAG
1683  002236  005037  001014                  CLR     PCNTR            ;CLEAR PASS COUNTER
1684  002242  004737  020620                  JSR     PC,GTSWR         ;GET SWITCHES
```

```
                                        ;TEST SCHEDULAR**********

002246  052777  000100  176316  TSCD:   BIS     #100,@TKS       ;SET KEYBOARD INTERRUPT ENABLE
002254  005737  000042                  TST     @042            ;ACT MODE ?
002260  001407                          BEQ     1$              ;BRANCH IF NOT
002262  032737  000004  172466          BIT     #4,@#172466
002270  001403                          BEQ     1$
002272  012737  000001  001010          MOV     #1,SLVTYP
002300  005037  000774          1$:     CLR     WPGFL           ;CLEAR WRAP PATRN FLAG
002304  005037  000734                  CLR     STFLG           ;CLEAR SINGLE TEST FLAG
002310  017700  176254                  MOV     @#SWR,R0
002314  042700  177700                  BIC     #177700,R0      ;BRANCH IF SINGLE
002320  001122                          BNE     STSCD           ;TEST SELECTED
002322  005737  001422                  TST     CHNFLG          ;;BRANCH IF NOT IN CHAIN MODE
(1)  002326  001457                          BEQ     TSCDA
(1)  002330  012737  177777  000520          MOV     #-1,DRVN        ;;INITIALIZE DRIVE #
(1)  002336  012737  177777  000660  NXTDRV: MOV     #-1,SLVN        ;;INITIALIZE SLAVE #
(1)  002344  012777  000040  176146  1$:     MOV     #40,@CS         ;;INIT CONTROLLER
(1)  002352  005237  000620                  INC     DRVN            ;;STEP DRIVE #
(1)  002356  022737  000010  000620          CMP     #10,DRVN        ;;EXIT IF ALL DRIVES TESTED
(1)  002364  001521                          BEQ     $DONE           ;;FOR AVAILABILITY
(1)  002366  013777  000620  176124          MOV     DRVN,@CS        ;;LOAD DRIVE #
(1)  002374  005777  176110                  TST     @CS             ;;ACCESS DRIVE
(1)  002400  032777  010000  176112          BIT     #10000,@CS      ;;BRANCH IF DRIVE NON EXISTANT
(1)  002406  001356                          BNE     1$              ;;(NED = 1)
(1)  002410  005237  000660          NXTSLV: INC     SLVN            ;;STEP SLAVE # AND BRANCH
(1)  002414  001011                          BNE     1$              ;;IF NOT SLAVE 0
(1)  002416  005737  000620                  TST     DRVN            ;;BRANCH IF NOT DRIVE # 0
(1)  002422  001006                          BNE     1$
(1)  002424  122737  000006  000041          CMPB    #6,@#41         ;;BRANCH IF NOT TMDP
(1)  002432  001002                          BNE     1$
(1)  002434  005237  000660                  INC     SLVN            ;;STEP TO SLAVE # 1
(1)  002440  022737  000010  000660  1$:     CMP     #10,SLVN        ;;BRANCH IF ALL SLAVES TESTED
(1)  002446  001733                          BEQ     NXTDRV          ;;FOR AVAILABILITY
(1)  002450  013777  000660  176064          MOV     SLVN,@TC        ;;LOAD SLAVE UNIT #
(1)  002456  032777  002000  176052          BIT     #2000,@DT       ;;BRANCH IF SLAVE NOT
(1)  002464  001751                          BEQ     NXTSLV          ;;PRESENT (SPR - 0)
1701  002466  012737  001056  000736  TSCDA:  MOV     #TSTTBL,LTADD
1702  002474  062737  000004  000736  TSCD0:  ADD     #4,LTADD
1703  002502  013737  000736  000710  TSCD1:  MOV     LTADD,ITRLP
1704  002510  062737  000002  000710          ADD     #2,ITRLP        ;SET ITERATION ADDRESS
1705  002516  005037  000614                  CLR     HDRFL           ;CLEAR PRINT HEADER FLAG
1706  002522  017700  176210                  MOV     @LTADD,R0       ;SET POINTER TO TEST
1707  002526  000110                          JMP     (R0)            ;GO TO TEST
1708  002530  032777  002000  176032  TSCD2:  BIT     #2000,@SWR      ;SEE IF HALT ON TEST
1709  002536  001403                          BEQ     TSCD3           ;IF NOT: BR
1710  002540  000000                          HALT
1711  002542  005037  000774                  CLR     WPGFL           ;CLEAR WRAP DATA GENERATOR FLAG
1712  002546  005737  000734          TSCD3:  TST     STFLG           ;SE IF SINGLE TEST
1713  002552  001750                          BEQ     TSCD0           ;IF NOT: BR
1714  002554  017700  176010                  MOV     @SWR,R0
1715  002560  042700  177700                  BIC     #177700,R0      ;BRANCH IF ALL TESTS DESIRED
1716  002564  001630                          BEQ     TSCD            ;IF SO: BR
1717  002566  012737  000001  000734  STSCD:  MOV     #1,STFLG        ;SET SINGLE TEST FLAG
1718  002574  023700  001330                  CMP     TLAST,R0        ;SEE IF EXCEEDED TESTS
```

```
1719  002600  002410                        BLT     TEND            ;IF SO: BR
1720  002602  006300                        ASL     R0
1721  002604  006100                        ROL     R0              ;SET TABLE MODIFIER
1722  002606  012737  001056  000736        MOV     #TSTTBL,LTADD
1723  002614  060037  000736                ADD     R0,LTADD        ;SET TEST POINTER
1724  002620  000730                        BR      TSCD1
1725  002622  005737  001422        TEND:   TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
1726  002626  001270                        BNE     NXTSLV          ;STEP TO NEXT SLAVE
1727  002630  012704  022563        $DONE:  MOV     #MSG41,R4
1728  002634  004737  017760                JSR     PC,TTOUT        ;PRINT END OF PASS
1729  002640  013703  001014                MOV     PCNTR,R3
1730  002644  004737  020172                JSR     PC,OCTP         ;PRINT PASS NUMBER
1731  002650  005000                        CLR     R0              ;DELAY WAITING FOR
1732  002652  005300                1$:     DEC     R0
1733  002654  001376                        BNE     1$
1734  002656  013700  000042                MOV     @#42,R0         ;GET ACT11 RETURN ADDRESS
 (1)  002662  001405                        BEQ     HERE            ;BRANCH IF NOT ACT11
 (1)  002664  000005                        RESET
 (1)  002666  004710                $ENDAD: JSR     PC,(R0)
 (1)  002670  000240                        NOP
 (1)  002672  000240                        NOP
 (1)  002674  000240                        NOP
 (1)  002676  000240                HERE:   NOP
1735  002700  005737  001422                TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
1736  002704  001005                        BNE     TENDX
1737  002706  032777  010000  175654        BIT     #10000,@SWR     ;SEE IF HALT ON PASS
1738  002714  001401                        BEQ     TENDX           ;IF NOT: BR
1739  002716  000000                        HALT
1740  002720  012737  000001  001012  TENDX: MOV     #1,SKAT         ;SET SKIP ADDRESS TEST FLAG
1741  002726  005237  001014                INC     PCNTR           ;BUMP PASS COUNTER
1742  002732  000137  002246                JMP     TSCD            ;RESTART
1743                                        ;LOGIC TEST 1: DRIVE ADDRESSING**********
1744
1745  002736  012737  024012  000616  LT1:  MOV     #MSLT1,EMADDR   ;**B SET ERROR MSG HDR ADDRESS
1746  002744  013737  000620  000674        MOV     DRVN,TEMP3      ;GET DRIVE # TO BE TESTED
1747  002752  013701  000620                MOV     DRVN,R1
1748  002756  005737  001012                TST     SKAT            ;SEE IF SKIP ADDRESS TESTS
1749  002762  001403                        BEQ     1$              ;IF NOT: BR
1750  002764  005737  000734                TST     STFLG           ;SEE IF SINGLE TEST
1751  002770  001506                        BEQ     LT1X            ;IF NOT: BR
1752  002772  032777  001000  175570  1$:   BIT     #1000,@SWR      ;BRANCH IF MAN INTERVENTION
1753  003000  001430                        BEQ     LT1A            ;NOT SELECTED
1754  003002  012704  021226        LT1GO:  MOV     #MSG2A,R4
1755  003006  004737  017200                JSR     PC,INST         ;PRINT TEST INSTRUCTIONS
1756  003012  012704  021165        LT1G:   MOV     #MSG2,R4
1757  003016  004737  017760                JSR     PC,TTOUT        ;REQUEST DRIVE NUMBER
1758  003022  012705  000674                MOV     #TEMP3,R5       ;TTR ROUTINE RETURNS RESPONSE TO (R5)
1759  003026  012701  000002                MOV     #2,R1
1760  003032  012702  000007                MOV     #7,R2
1761  003036  012703  000000                MOV     #0,R3
1762  003042  004737  017436                JSR     PC,TTR          ;GET DRIVE NUMBER
1763  003046  005737  000670                TST     TEMP1           ;SEE IF ANOTHER DRIVE
1764  003052  001455                        BEQ     LT1X            ;IF NOT: BR
1765  003054  005001                        CLR     R1              ;SELECT DRIVE 0
1766  003056  012700  000010                MOV     #10,R0          ;SET NUMBER OF DRIVES
1767  003062  012777  000040  175430  LT1A: MOV     #40,@CS         ;INIT
```

```
1768  003070  010177  175424          MOV     R1,@CS          ;SELECT DRIVE
1769  003074  005777  175410          TST     @C1             ;ACCESS DRIVE
1770  003100  032777  010000  175412  BIT     #10000,@CS      ;SEE IF NED
1771  003106  001010                  BNE     LT1B            ;IF SO: BR
1772  003110  032777  001000  175452  BIT     #1000,@SWR      ;BRANCH IF NOT MANUAL INTERVENTION
1773  003116  001433                  BEQ     LT1X
1774  003120  023701  000674          CMP     TEMP3,R1        ;SEE IF SHOULD BE NED
1775  003124  001404                  BEQ     LT1C            ;IF NOT: BR
1776  003126  000407                  BR      LT1ER           ;ELSE GO TO ERROR
1777  003130  023701  000674  LT1B:   CMP     TEMP3,R1        ;SEE IF SHOULD BE NED
1778  003134  001410                  BEQ     LT1ER1
1779  003136  005300          LT1C:   DEC     R0
1780  003140  001724                  BEQ     LT1G            ;IF DONE ALL: BR
1781  003142  005201                  INC     R1              ;SELECT NEXT DRIVE
1782  003144  000746                  BR      LT1A            ;CONTINUE
1783  003146  012737  000001  000712  LT1ER:  MOV     #1,EXFL         ;FLAG EXPT
1784  003154  000403                  BR      LT1ER2
1785  003156  012737  000002  000712  LT1ER1: MOV     #2,EXFL         ;FLAG NOT EXPT
1786  003164  012737  021555  000666  LT1ER2: MOV     #MSG3,ERADD     ;FLAG CONDITION
1787  003172  012737  003062  000706          MOV     #LT1A,SCOLP     ;SET SCOPE ADDRESS
1788  003200  004737  015230                  JSR     PC,LTGER        ;GO PRINT LOGIC TEST ERROR
1789  003204  000754                  BR      LT1C            ;CONTINUE TEST
1790  003206  000137  002530          LT1X:   JMP     TSCD2           ;RETURN TO SCHED
1791
```

```
                                                ;LOGIC TEST 2: REGISTER ADDRESSING**********

003212  000240                          LT2:    NOP
003214  012777  000040  175276          LT2IT:  MOV     #40,@CS                 ;INIT
003222  013777  000620  175270                  MOV     DRVN,@CS                ;SELECT DRIVE
003230  012737  024066  000616                  MOV     #MSLT2,EMADDR           ;SAVE LT2 HEADER ADDRESS
003236  012705  000510                          MOV     #C1,R5                  ;SET ADDRESS OF FIRST REGISTER
003242  012700  000016                          MOV     #16,R0                  ;SET NUMBER OF REGISTERS
003246  012702  000622                          MOV     #TR00,R2                ;SET START OF REGISTER BUFFER
003252  011501                          LT2A:   MOV     (R5),R1
003254  011112                                  MOV     (R1),(R2)               ;READ REGISTER
003256  032777  020000  175224                  BIT     #20000,@C1              ;SEE IF ERROR
003264  001402                                  BEQ     LT2B                    ;IF NOT: BR
003266  004737  003316                          JSR     PC,LT2ER1               ;ELSE GO TO ERROR 1
003272  032777  000002  175224          LT2B:   BIT     #2,@ER                  ;SEE IF ILR
003300  001402                                  BEQ     LT2C                    ;IF NOT: BR
003302  004737  003334                          JSR     PC,LT2ER2               ;ELSE GO TO ERROR 2
003306  022225                          LT2C:   CMP     (R2)+,(R5)+             ;BUMP ADDRESS
003310  005300                                  DEC     R0
003312  001357                                  BNE     LT2A                    ;CONTINUE FOR ALL REGISTERS
003314  000434                                  BR      LT2X

003316  012737  000002  000712          LT2ER1: MOV     #2,EXFL                 ;FLAG NOT EXPECTED
003324  012737  021377  000666                  MOV     #MSG4,ERADD             ;POINT TO CONTROLLER ERROR
003332  000415                                  BR      LT2ERG                  ;GO TO ERROR
003334  012737  000002  000712          LT2ER2: MOV     #2,EXFL                 ;FLAG NOT EXPECTED
003342  012737  021415  000666                  MOV     #MSG5,ERADD             ;POINT TO DRIVE ERROR
003350  000406                                  BR      LT2ERG                  ;GO TO ERROR
003352  012737  000001  000712          LT2ER3: MOV     #1,EXFL                 ;FLAG EXPECTED
003360  012737  021377  000666                  MOV     #MSG4,ERADD             ;POINT TO DRIVE
003366  012737  003402  000706          LT2ERG: MOV     #LT2LP,SCOLP            ;SET SCOPE ADDRESS
003374  004737  015230                          JSR     PC,LTGER                ;GO PRINT
003400  000207                                  RTS     PC                      ;ELSE CONTINUE
003402  005726                          LT2LP:  TST     (SP)+                   ;RESET STACK
003404  000722                                  BR      LT2A                    ;LOOP
003406  004737  016600                  LT2X:   JSR     PC,ITER                 ;GO SEE IF ITERATIONS
003412  000137  002530                          JMP     TSCD2                   ;RETURN TO SCHED
```

SEQ 0044

```
1831                                            ;LOGIC TEST 3: CONTROL BUS**********
1832
1833  003416  000240                      LT3:    NOP
1834  003420  012737  024145  000616      LT3IT:  MOV     #MSLT3,EMADDR   ;SET TEST HEADER
1835  003426  012701  000001                      MOV     #1,R1           ;PRESET PATTERN 1
1836  003432  012700  000020                      MOV     #20.R0          ;SET PATTERN CHANGE NUMBER
1837  003436  004737  016726              LT3A:   JSR     PC,INIT1        ;GO INIT
1838  003442  010177  175050                      MOV     R1,@FC          ;WRITE TO FC
1839  003446  032777  000010  175050              BIT     #10,@ER         ;SEE IF CPAR (TM03)
1840  003454  001012                              BNE     LT3ER1          ;IF SO: BR
1841  003456  017702  175034              LT3B:   MOV     @FC,R2          ;READ FC
1842  003462  032777  020000  175020              BIT     #20000,@C1      ;SEE IF MCPE (RH)
1843  003470  001017                              BNE     LT3ER2          ;IF SO: BR
1844  003472  005300                      LT3C:   DEC     R0              ;SEE IF DONE PATTERN CHANGES
1845  003474  001426                              BEQ     LT3X            ;IF SO: BR
1846  003476  006301                              ASL     R1              ;CHANGE PATTERN
1847  003500  000756                              BR      LT3A            ;CONTINUE
1848  003502  012737  021730  000666  LT3ER1:  MOV     #MSG11,ERADD    ;SET ERROR CODE
1849  003510  012737  003436  000706              MOV     #LT3A,SCOLP     ;SET SCOPE ADDRESS
1850  003516  017702  174774                      MOV     @FC,R2          ;GET DATA
1851  003522  004737  016336                      JSR     PC,LTGER1       ;GO DO ERROR
1852  003526  000753                              BR      LT3B
1853  003530  012737  021704  000666  LT3ER2:  MOV     #MSG10,ERADD    ;SET ERROR CODE
1854  003536  012737  003456  000706              MOV     #LT3B,SCOLP     ;SET SCOPE ADDRESS
1855  003544  004737  016336                      JSR     PC,LTGER1       ;GO DO ERROR
1856  003550  000750                              BR      LT3C
1857  003552  105701                      LT3X:   TSTB    R1              ;SEE IF DONE PATTERN 2
1858  003554  100405                              BMI     LT3XX           ;IF SO: BR
1859  003556  012701  000401                      MOV     #401,R1         ;SET PATTERN 2
1860  003562  012700  000010                      MOV     #10,R0          ;SET PATTERN CHANGE NUMBER
1861  003566  000723                              BR      LT3A            ;DO PATTERN 2
1862  003570  004737  016600              LT3XX:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
1863  003574  000137  002530                      JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
1865
1866                                        ;LOGIC TEST 4: SLAVE ADDRESSING**********
1867
1868  003600  013737  000660  000674  LT4:  MOV   SLVN,TEMP3
1869  003606  013701  000660              MOV   SLVN,R1
1870  003612  005737  001012              TST   SKAT          ;SEE IF SKIP ADDRESS TESTS
1871  003616  001403                      BEQ   1$            ;IF NOT: BR
1872  003620  005737  000734              TST   STFLG         ;SEE IF SINGLE TEST
1873  003624  001564                      BEQ   LT4X          ;IF NOT: BR
1874  003626  032777  001000  174734  1$: BIT   #1000,@SWR    ;BRANCH IF MAN INTERVETION
1875  003634  001430                      BEQ   LT4A          ;NOT SELECTED
1876  003636  012704  021533          LT4G0: MOV  #MSG8A,R4
1877  003642  004737  017200              JSR   PC,INST       ;PRINT TEST INSTRUCTIONS
1878  003646  012704  021472          LT4G: MOV   #MSG8,R4
1879  003652  004737  017760              JSR   PC,TTOUT      ;REQUEST SLAVE
1880  003656  012705  000674              MOV   #TEMP3,R5
1881  003662  012701  000002              MOV   #2,R1
1882  003666  012702  000007              MOV   #7,R2
1883  003672  012703  000000              MOV   #0,R3
1884  003676  004737  017436              JSR   PC,TTR        ;GET SLAVE NUMBER
1885  003702  005737  000670              TST   TEMP1         ;SEE IF SLAVE
1886  003706  001533                      BEQ   LT4X          ;IF NOT: BR
1887  003710  005001                      CLR   R1            ;SELECT SLAVE 0
1888  003712  012700  000010              MOV   #10,R0        ;SET NUMBER OF SLAVES
1889  003716  012777  000040  174574  LT4A: MOV  #40,@CS      ;INIT
1890  003724  013777  000620  174566      MOV   DRVN,@CS      ;SELECT DRIVE
1891  003732  010177  174604              MOV   R1,@TC        ;SELECT SLAVE
1892  003736  017703  174574              MOV   @DT,R3        ;GET DT
1893  003742  020137  000674              CMP   R1,TEMP3      ;SEE IF SHOULD HAVE SPR
1894  003746  001404                      BEQ   LT4B          ;IF SO: BR
1895  003750  032703  002000              BIT   #2000,R3      ;SEE IF SPR
1896  003754  001461                      BEQ   LT4D          ;IF NOT: BR
1897  003756  000464                      BR    LT4ER1        ;GO TO ERROR 1
1898  003760  032703  002000          LT4B: BIT   #2000,R3    ;SEE IF NO SLAVE PRESENT
1899  003764  001465                      BEQ   LT4ER2        ;(SPR=0)
1900  003766  012704  023642          LT4C: MOV   #MSG64,R4   ;TYPE 'SLAVE TYPE - '
1901  003772  004737  017760              JSR   PC,TTOUT
1902  003776  012702  000001              MOV   #1,R2         ;PRESET SLAVE TYPE = TU77
1903  004002  012704  023660              MOV   #MSG65,R4     ;SET UP TO TYPE 'TU77'
1904  004006  022777  142054  174522      CMP   #142054,@DT   ;BRANCH IF TU77
1905  004014  001[illegible]              BEQ   1$
1906  004016  012704  023665              MOV   #MSG66,R4     ;CHANGE SLAVE TYPE TO 'TE16'
1907  004022  005302                      DEC   R2            ;CHANGE SLAVE TYPE TO TE16
1908  004024  022777  142051  174504      CMP   #142051,@DT   ;BRANCH IF TE16
1909  004032  001403                      BEQ   1$
1910  004034  005302                      DEC   R2            ;CHENGE SLAVE TYPE TO ILLEGAL
1911  004036  012704  024002              MOV   #MSG69,R4
1912  004042  004737  017760          1$: JSR   PC,TTOUT      ;TYPE SLAVE TYPE (TU77,TE16, OR ILLEGAL)
1913  004046  020237  001010              CMP   R2,SLVTYP     ;BRANCH IF HARDWARE SLAVE TYPE IS THE
1914  004052  001406                      BEQ   4$            ;SAME AS USER SPACIFIED SLAVE TYPE
1915  004054  012704  023730              MOV   #MSG68,R4     ;++B TYPE 'INCORRECT SLAVE TYPE'
1916  004060  004737  017760              JSR   PC,TTOUT      ;++B
1917  004064  000137  002622              JMP   TEND          ;++B EXIT TEST
1918  004070  012704  022405          4$: MOV   #MSG30,R4
1919  004074  004737  017760              JSR   PC,TTOUT      ;PRINT SERIAL NUMBER TAG
1920  004100  017703  174434              MOV   @SN,R3
```

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

```
1939                                        ;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST**********
1940
1941  004202  012737  004314  000616  LT5:    MOV     #MSG15,EMADDR   ;SET TEST HEADER
1942  004210  004737  016726          LT5IT:  JSR     PC,INIT1        ;GO INIT
1943  004214  012700  000032                  MOV     #32,R0          ;SET LOOP FOR BITS 4-0
1944  004220  005001                          CLR     R1              ;SET TEST WORD
1945  004222  010177  174306          LT5A:   MOV     R1,@MR          ;SEND TEST WORD TO MR
1946  004226  017702  174302                  MOV     @MR,R2          ;READ MR
1947  004232  042702  177740                  BIC     #177740,R2      ;MASK BITS 4-0
1948  004236  020102                          CMP     R1,R2           ;SEE IF EXPT = RECVD
1949  004240  001026                          BNE     LT5ER1
1950  004242  005300                  LT5B:   DEC     R0
1951  004244  001402                          BEQ     LT5C            ;IF DONE LOOP: BR
1952  004246  005201                          INC     R1              ;BUMP TEST WORD
1953  004250  000764                          BR      LT5A            ;CONTINUE LOOP
1954  004252  012701  000015          LT5C:   MOV     #15,R1          ;SET TEST WORD = WAM 3
1955  004256  012700  001000                  MOV     #1000,R0        ;SET LOOP FOR BITS 15-7
1956  004262  010177  174246          LT5D:   MOV     R1,@MR          ;LOAD MR
1957  004266  017702  174242                  MOV     @MR,R2          ;READ MR
1958  004272  042702  000140                  BIC     #140,R2         ;MASK OUT BITS 5,6
1959  004276  020102                          CMP     R1,R2           ;SEE IF EXPT = RECVD
1960  004300  001401                          BEQ     LT5E            ;IF SO: BR
1961  004302  000416                          BR      LT5ER2          ;ELSE GO TO ERR 2
1962  004304  005300                  LT5E:   DEC     R0
1963  004306  001425                          BF      LT5X            ;IF DONE LOOP: BR
1964  004310  062701  000200                  ...     #200,R1         ;BUMP TEST WORD
1965  004314  000762                          ...     LT5D            ;CONTINUE LOOP
1966  004316  012737  021773  000666  LT5ER1:  ...     #MSG14,ERADD    ;SET ERROR CODE
1967  004324  012737  004222  000706          ...     #LT5A,SCOLP     ;SET SCOPE ADDRESS
1968  004332  004737  016336                  JSR     PC,LTGER1       ;GO TO ERROR
1969  004336  000741                          BR      LT5B            ;CONTINUE
1970  004340  012737  022010  000666  LT5ER2:  MOV     #MSG15,ERADD    ;SET ERROR CODE
1971  004346  012737  004262  000706          MOV     #LT5D,SCOLP     ;SET SCOPE ADDRESS
1972  004354  004737  016336                  JSR     PC,LTGER1       ;GO TO ERROR
1973  004360  000751                          BR      LT5E            ;CONTINUE
1974  004362  004737  016600          LT5X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
1975  004366  000137  002530                  JMP     TSCD2           ;RETURN TO SCHED
1976
```

```
                                                  ;LOGIC TEST 6: TC REGISTER BIT TEST**********

004372  000240                    LT6:     NOP
004374  012737  004363  000616    LT6IT:   MOV     #MSLT6,EMADDR     ;POINT TO LT6 HEADER
004402  012700  000003                     MOV     #3,R0             ;SET NUMBER OF TESTS
004406  005001                    LT6A1:   CLR     R1
004410  004737  016726            LT6A:    JSR     PC,INIT1          ;GO INIT
004414  010177  174122            LT6B:    MOV     R1,@TC            ;WRITE TC
004420  017702  174116                     MOV     @TC,R2            ;READ TC
004424  042702  160000                     BIC     #160000,R2        ;MASK OUT SAC
004430  020102                             CMP     R1,R2             ;SEE IF EXPT = RECVD
004432  001010                             BNE     LT6ER1            ;IF NOT: BR
004434  005300                    LT6D:    DEC     R0
004436  001417                             BEQ     LT6X              ;IF DONE ALL: BR
004440  020027  000001                     CMP     #1,R0             ;SEE IF RESET TEST
004444  001760                             BEQ     LT6A1             ;IF SO: BR
004446  012701  017777                     MOV     #17777,R1         ;SET TEST WORD
004452  000756                             BR      LT6A              ;DO SET TEST
004454  012737  022036  000666    LT6ER1:  MOV     #MSG18,ERADD      ;SET ERROR CODE
004462  012737  004414  000706             MOV     #LT6B,SCOLP       ;SET SCOPE ADDRES
004470  004737  016336                     JSR     PC,LTGER1         ;GO TO ERROR
004474  000757                             BR      LT6D              ;CONTINUE
004476  004737  016600            LT6X:    JSR     PC,ITER           ;GO SEE IF ITERATIONS
004502  000137  002530                     JMP     TSCD2             ;RETURN TO SCHED
```

```
                                        ;LOGIC TEST 7: FRAME COUNT BIT TEST**********

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

```

```
                                          ;LOGIC TEST 10: FUNCTION CODE BIT TEST**********

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

```
2056
2057                                        ;LOGIC TEST 11: GO BIT SET RESET**********
2058
2059  004746  000240                LT11:   NOP
2060  004750  012737  024557  000616 LT11IT: MOV     #MSLT11,EMADDR  ;SET TEST HEADER
2061  004756  004737  016726              JSR     PC,INIT1        ;GO INIT
2062  004762  017702  173522              MOV     @C1,R2          ;READ C1
2063  004766  032702  000001              BIT     #1,R2           ;SEE IF GO=0
2064  004772  001030                      BNE     LT11E1
2065  004774  012777  000015  173532 LT11B: MOV     #15,@MR         ;SELECT WAM 3
2066  005002  005077  173510              CLR     @FC             ;ASSURE FCS = 1
2067  005006  052777  001700  173526      BIS     #1700,@TC       ;ASSURE FMT OK
2068  005014  012777  000071  173466      MOV     #71,@C1         ;SET READ+GO
2069  005022  017702  173462              MOV     @C1,R2          ;READ C1
2070  005026  032702  000001              BIT     #1,R2           ;SEE IF GO =1
2071  005032  001424                      BEQ     LT11E2
2072  005034  004737  016726      LT11C:  JSR     PC,INIT1        ;GO INIT
2073  005040  017702  173444              MOV     @C1,R2          ;READ C1
2074  005044  032702  000001              BIT     #1,R2           ;SEE IF GO=0
2075  005050  001444                      BEQ     LT11X           ;IF SO-BR
2076  005052  000430                      BR      LT11E3          ;ELSE GO TO ERROR 3
2077  005054  012737  022126  000666 LT11E1: MOV     #MSG21,ERADD    ;SET ERROR CODE
2078  005062  012702  000001              MOV     #1,R2           ;SET REVD
2079  005066  005001                      CLR     R1              ;SET EXPT
2080  005070  012737  004750  000706      MOV     #LT11IT,SCOLP           ;SET SCOPE ADDRESS
2081  005076  004737  016336              JSR     PC,LTGER1       ;GO PRINT ERROR
2082  005102  000734                      BR      LT11B           ;ELSE CONTINUE
2083  005104  012737  022164  000666 LT11E2: MOV     #MSG22,ERADD    ;SET ERROR CODE
2084  005112  005002                      CLR     R2              ;SET RCVD
2085  005114  012701  000001              MOV     #1,R1           ;SET EXPT
2086  005120  012737  004774  000706      MOV     #LT11B,SCOLP    ;SET SCOPE ADDRESS
2087  005126  004737  016336              JSR     PC,LTGER1       ;GO PRINT ERROR
2088  005132  000740                      BR      LT11C           ;ELSE CONTINUE
2089  005134  012737  022205  000666 LT11E3: MOV     #MSG23,ERADD    ;SET ERROR CODE
2090  005142  005001                      CLR     R1              ;SET EXPT
2091  005144  012702  000001              MOV     #1,R2           ;SET RCVD
2092  005150  012737  005034  000706      MOV     #LT11C,SCOLP    ;SET SCOPE ADDRESS
2093  005156  004737  016336              JSR     PC,LTGER1       ;GO PRINT ERROR
2094  005162  004737  016600      LT11X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2095  005166  000137  002530              JMP     TSCD2           ;RETURN TO SCHED
```

```
2097
2098                                        ;LOGIC TEST 12: DRIVE READY BIT**********
2099
2100  005172  000240                LT12:   NOP
2101  005174  012737  024624  000616  LT12IT: MOV     #MSLT12,EMADDR  ;SET TEST HEADER
2102  005202  004737  016726          JSR     PC,INIT1        ;GO INIT
2103  005206  032777  000200  173306  BIT     #200,@DS        ;SEE IF DRY=1
2104  005214  001426                  BEQ     LT12E1
2105  005216  012777  000015  173310  LT12B:  MOV     #15,@MR         ;SET WAM3
2106  005224  005077  173266          CLR     @FC             ;ASSURE FCS = 1
2107  005230  052777  001700  173304  BIS     #1700,@TC       ;ASSURE FMT OK
2108  005236  012777  000071  173244  MOV     #71,@C1         ;SET READ+GO
2109  005244  032777  000200  173250  BIT     #200,@DS        ;SEE IF DRY=0
2110  005252  001020                  BNE     LT12E2
2111  005254  004737  016726  LT12C:  JSR     PC,INIT1        ;GO INIT
2112  005260  032777  000200  173234  BIT     #200,@DS        ;SEE IF DRY=1
2113  005266  001033                  BNE     LT12X           ;IF SO: BR
2114  005270  000422                  BR      LT12E3          ;ELSE GO TO ERROR 3
2115  005272  012737  022240  000666  LT12E1: MOV     #MSG24,ERADD    ;SET ERROR CODE
2116  005300  012737  005174  000706  MOV     #LT12IT,SCOLP   ;SET SCOPE ADDRESS
2117  005306  004737  016330          JSR     PC,LTGER2       ;GO TO ERROR
2118  005312  000741                  BR      LT12B           ;CONTINUE
2119  005314  012737  022266  000666  LT12E2: MOV     #MSG25,ERADD    ;SET ERROR CODE
2120  005322  012737  005216  000706  MOV     #LT12B,SCOLP    ;SET LOOP ADDRESS
2121  005330  004737  016330          JSR     PC,LTGER2       ;GO PRINT ERROR
2122  005334  000747                  BR      LT12C           ;CONTINUE
2123  005336  012737  022315  000666  LT12E3: MOV     #MSG25A,ERADD   ;SET ERROR CODE
2124  005344  012737  005254  000706  MOV     #LT12C,SCOLP    ;SET ERROR LOOP
2125  005352  004737  016330          JSR     PC,LTGER2       ;GET PRINT ERROR
2126  005356  004737  016600  LT12X:  JSR     PC,ITER         ;GO TO ITERATION SUBROUTINE
2127  005362  000137  002530          JMP     TSCD2           ;RETURN TO SCHED
```

```
2129
2130                                            ;LOGIC TEST 13: INTERRUPT TEST**********
2131
2132  005366  005000                    LT13:   CLR     R0
2133  005370  012737  024675  000616            MOV     @#LT13,EMADDR   ;SET TEST HEADER
2134  005376  004737  016726            LT13IT: JSR     PC,INIT1        ;GO INIT,SELECT DRIVE, SELECT ABOVE
2135  005402  012737  005460  000664            MOV     @#LT13X,RTRN    ;SET RETURN ADDRESS
2136  005410  005077  173074                    CLR     @#C1            ;CLEAR CS1
2137  005414  005077  173146                    CLR     @#PSW           ;SET PRIORITY
2138  005420  052777  000100  173062            BIS     #100,@#C1       ;BIT SET IE
2139  005426  005300                    LT13A:  DEC     R0
2140  005430  001376                            BNE     LT13A           ;AWAIT INTERRUPT
2141  005432  012777  000340  173136    LT13E1: MOV     #340,@#PSW      ;RESET PRIORITY
2142  005440  012737  022342  000666            MOV     #MSG26,ERADD    ;SET ERROR CODE
2143  005446  012737  005376  000706            MOV     #LT13IT,SCOLP   ;SET LOOP ADDRESS
2144  005454  004737  016330                    JSR     PC,LTGER2       ;GO PRINT ERROR
2145  005460  004737  016600            LT13X:  JSR     PC,ITER         ;GO TO ITERATION SUBROUTINE
2146  005464  000137  002530                    JMP     @#SCD2          ;RETURN TO SCHED
```

```
                                                ;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**********

005470  032737  001000  173072  LT14:    BIT     #1000,@SWR        ;SEE IF INHIB MAN TST
005476  001005                           BNE     LT14A             ;IF NOT: BR
005500  005737  000734                   TST     STFLG             ;SEE IF SINGLE TEST
005504  001433                           BEQ     LT14XX            ;IF NOT: BR
005506  000137  016650                   JMP     INMT              ;ELSE GO PRINT INHIB MSG
005512  012737  024742  000616  LT14A:   MOV     #MSLT14,EMADDR    ;SET TEST HEADER
005520  012704  027147                   MOV     #MMSG1,R4         ;SET INSTRUCTION ONE
005524  004737  017200                   JSR     PC,INST           ;GO DO INSTRUCTION
005530  004737  016726          LT14IT:  JSR     PC,INIT1          ;INIT, SELECT DRIVE + SLAVE
005534  012701  014602                   MOV     #14602,R1         ;SET TEST WORD
005540  017702  172756                   MOV     @DS,R2            ;ASSURE MOL,WRL,DPR,DRY,BOT
005544  020102                           CMP     R1,R2
005546  001410                           BEQ     LT14X             ;IF SO: BR
005550  012737  005530  000706           MOV     #LT14IT,SCOLP     ;SET LOOP ADDRESS
005556  012737  022371  000666           MOV     #MSG27,ERADD      ;SET ERROR CODE
005564  004737  016336                   JSR     PC,LTGER1         ;GO PRINT ERROR
005570  004737  016600          LT14X:   JSR     PC,ITER           ;GO SEE IF ITERATION
005574  000137  002530          LT14XX:  JMP     TSCD2             ;RETURN TO SCHED
```

```
                                        ;LOGIC TEST 15: STATUS AT BOT, OFFLINE, LOADED, NO WRITE RING**********
005600  032777  001000  172762  LT15:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
005606  001005                          BNE     LT15A           ;IF NOT: BR
005610  005737  000734                  TST     STFLG           ;SEE IF SINGLE TEST
005614  001433                          BEQ     LT15XX          ;IF NOT: BR
005616  000137  016650                  JMP     INMT            ;ELSE GO PRINT INHIB MSG
005622  012737  025051  000616  LT15A:  MOV     #MSLT15,EMADDR  ;SET TEST HEADER
005630  012704  027245                  MOV     #MMSG2,R4
005634  004737  017200                  JSR     PC,INST         ;PRINT INSTRUCTION
005640  004737  016736          LT15IT: JSR     PC,INIT2        ;GO INIT, SELECT DRIVE, SLAV
005644  012701  100700                  MOV     #100700,R1      ;SET TEST WORD
005650  017702  172646                  MOV     @DS,R2          ;READ STATUS
005654  020102                          CMP     R1,R2           ;SEE OF EXPT=RCVD
005656  001410                          BEQ     LT15X
005660  012737  005640  000706          MOV     #LT15IT,SCOLP   ;SET LOOP ADDRESS
005666  012737  022371  000666          MOV     #MSG27,ERADD    ;SET ERROR CODE
005674  004737  016336                  JSR     PC,LTGER1       ;GO PRINT ERROR
005700  004737  016600          LT15X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
005704  000137  002530          LT15XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2197
2198                                        ;LOGIC TEST 16: STATUS AT EOT, ON LINE, NO WRITE RING**********
2199
2200  005710  032777  001000  172652  LT16:   BIT     #1000,@#SWR     ;SEE IF INHIB MAN TST
2201  005716  001005                          BNE     LT16A           ;IF NOT: BR
2202  005720  005737  000734                  TST     STFLG           ;SEE IF SINGLE TEST
2203  005724  001433                          BEQ     LT16XX          ;IF NOT: BR
2204  005726  000137  016650                  JMP     INHT            ;ELSE GO PRINT INHIB MSG
2205  005732  012737  025141  000616  LT16A:  MOV     #MSLT16,EMADDR  ;SET TEST HEADER
2206  005740  012704  027266                  MOV     #MMSG3,R4
2207  005744  004737  017200                  JSR     PC,INST         ;GO PRINT INSTRUCTION
2208  005750  004737  016736          LT16IT: JSR     PC,INIT2        ;SELECT DRIVE,SLAVE
2209  005754  013701  006020                  MOV     STWD16,R1       ;SET TEST WORD (ATA!MOL!WRL!EOT!DPR!DRY!SSC!SLA)
2210  005760  017702  172536                  MOV     @DS,R2          ;READ STATUS
2211  005764  020102                          CMP     R1,R2           ;SEE IF EXPT=RCVD
2212  005766  001410                          BEQ     LT16X           ;IF SO: BR
2213  005770  012737  005750  000706          MOV     #LT16IT,SCOLP   ;SET LOOP ADDRESS
2214  005776  012737  022371  000666          MOV     #MSG27,ERADD    ;SET ERROR CODE
2215  006004  004737  016336                  JSR     PC,LTGER1       ;GO PRINT ERROR
2216  006010  004737  016600          LT16X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2217  006014  000137  002530          LT16XX: JMP     TSCD2           ;RETURN TO SCHED
2218  006020  116701                  STWD16: .WORD   116701          ;SET UP TEST WORD FOR TE16
2219                                                                  ;CONTENTS OF TEST WORD FOR
2220                                                                  ;A TU77 IS 116741
2221
```

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

```
2248                                    ;THE FOLLOWING 11 TESTS WILL TEST ALL POSSIBLE ERROR BITS
2249                                    ;BY FORCING THEIR CONDITIONS THROUGH VARIOUS ILLEGAL PROGRAMMING
2250                                    ;SEQUENCES AND USING THE MAINTENANCE WILL MODES AVAILABLE WITH TM03
2251                                    ;FOR EACH ERROR CONDITION SET THE APPROPRIATE STATUS WILL BE
2252                                    ;CHECKED. IE: ERR, ATA, SLA, SC ETC.
2253
2254                                    ;LOGIC TEST 20: ILLEGAL FUNCTION (ILF)**********
2255
2256  006134  012737  025311  000616  LT20:   MOV     #MSLT20,EMADDR  ;SET TEST HEADER
2257  006142  012737  006162  000706          MOV     #LT20A,SCOLP    ;SET LOOP ADDRESS
2258  006150  012700  000022          LT20IT: MOV     #22,R0          ;SET NUMBER OF ILL CODES
2259  006154  012737  000544  000670          MOV     #ILFT,TEMP1     ;POINT TO START IF TABLE
2260  006162  004737  016726          LT20A:  JSR     PC,INIT1        ;GO INIT, SELECT SLAVE + DRIVE
2261  006166  012777  177777  172316          MOV     #-1,@WC         ;SET WC= -1
2262  006174  012701  000001                  MOV     #1,R1           ;SET TEST WORD
2263  006200  117777  172464  172302          MOVB    @TEMP1,@C1      ;SET ILL CODE
2264  006206  017702  172312                  MOV     @ER,R2          ;READ ER
2265  006212  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2266  006214  001011                          BNE     LT20B           ;IF SO: BR
2267  006216  012737  027653  000666          MOV     #TMS17,ERADD    ;SET ERROR CODE
2268  006224  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
2269  006232  004737  015222                  JSR     PC,LTGER0       ;GO PRINT ERROR
2270  006236  000404                          BR      LT20C
2271  006240  020102          LT20B:  CMP     R1,R2           ;SEE UNEXPECTED ERRORS
2272  006242  001402                          BEQ     LT20C           ;IF NOT: BR
2273  006244  004737  015210                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2274  006250  005300          LT20C:  DEC     R0              ;SEE IF DONE ALL ILL CODES
2275  006252  001403                          BEQ     LT20X           ;IF SO: BR
2276  006254  005237  000670                  INC     TEMP1           ;BUMP ADDRESS
2277  006260  000740                          BR      LT20A           ;CONTINUE
2278  006262  004737  016500  LT20X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2279  006266  004737  015670          JSR     PC,DRVCLR
2280  006272  000137  002530          JMP     TSCD2           ;RETURN TO SCHED
```

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

```
2312
2313                                        ;LOGIC TEST 22: CONTROL BUS PARITY (CPAR)**********
2314
2315  006442  012737  025424  000616  LT22:   MOV     #MSLT22,EMADDR  ;SET TEST HEADER
2316  006450  012737  006456  000706          MOV     #LT22IT,SCOLP   ;SET SCOPE LOOP ADDRESS
2317  006456  004737  016726          LT22IT: JSR     PC,INIT1        ;INIT, SELECT SLAVE+DRIVE
2318  006462  052777  000020  172030          BIS     #20,@CS         ;ENABLE EVEN PARITY ON MB
2319  006470  012777  177777  172020          MOV     #-1,@FC         ;WRITE TO FC
2320  006476  012701  000010                  MOV     #10,R1          ;SET TEST WORD
2321  006502  042777  000020  172010          BIC     #20,@CS         ;RESET PARITY TO ODD
2322  006510  017702  172010                  MOV     @ER,R2          ;GET ER
2323  006514  030102                          BIT     R1,R2           ;SEE IF EXPT+RCVD
2324  006516  001011                          BNE     LT22A           ;IF SO: BR
2325  006520  012737  027675  000666          MOV     #TMS20,ERADD    ;SET ERROR CODE
2326  006526  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
2327  006534  004737  015222                  JSR     PC,LTGER0       ;GO PRINT ERROR
2328  006540  000404                          BR      LT22X
2329  006542  020102          LT22A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2330  006544  001402                          BEQ     LT22X           ;IF NOT: BR
2331  006546  004737  015210                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2332  006552  004737  016600          LT22X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2333  006556  004737  015670                  JSR     PC,DRVCLR
2334  006562  000137  002530                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2336
2337                                        ;LOGIC TEST 23: FORMAT ERROR(FMT)**********
2338
2339  006566  012737  025461  000616  LT23:   MOV     #MSLT23,EMADDR  ;SET TEST HEADER
2340  006574  012737  006602  000706          MOV     #LT23IT,SCOLP   ;SET SCOPE ADDRESS
2341  006602  004737  016726          LT23IT: JSR     PC,INIT1        ;GO INIT SELECT DRIVE+SLAVE
2342                                  ;++B    BIC     #360,@TC        ;++B DELETED SET ILLEGAL FORMAT
2343  006606  052777  000360  171726          BIS     #360,@TC        ;++B SET ILLEGAL FORMAT FOR BOTH M8906 & M8915
2344  006614  012701  000020                  MOV     #20,R1          ;SET TEST WORD
2345  006620  012777  000015  171706          MOV     #15,@MR         ;SET WAM 3
2346  006626  012777  000071  171654          MOV     #71,@C1         ;SET READ+GO
2347  006634  017702  171664                  MOV     @ER,R2          ;READ ER
2348  006640  030102                          BIT     R1,R2           ;SEE IF EXPT+RCVD
2349  006642  001011                          BNE     LT23A           ;IF SO: BR
2350  006644  012737  027704  000666          MOV     #TMS21,ERADD    ;SET ERROR CODE
2351  006652  012737  000001  000712          MOV     #1,EXFL         ;SET EXPT FLG
2352  006660  004737  015222                  JSR     PC,LTGERO       ;GO PRINT ERROR
2353  006664  000404                          BR      LT23X
2354  006666  020102          LT23A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2355  006670  001402                          BEQ     LT23X           ;IF NOT: BR
2356  006672  004737  015210                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2357  006676  004737  016600          LT23X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2358  006702  004737  015054                  JSR     PC,EORPA
2359  006706  004737  015670                  JSR     PC,DRVCLR
2360  006712  000137  002530                  JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                ;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)**********
                                                
006716  012737  025526  000616  LT24:   MOV     #MSLT24,EMADDR  ;SET TEST HEADER
006724  012737  006732  000706          MOV     #LT24IT,SCOLP   ;SET SCOPE ADDRESS
006732  012737  000005  000602  LT24IT: MOV     #5,ITAMI
006740  004737  016754                  JSR     PC,INIT3        ;GO INIT, SELECT DRIVE+SLAVE
006744  052777  000300  171570          BIS     #300,@TC        ;SET NORMAL FORMAT
006752  012777  030204  171534          MOV     #WDATA,@BA      ;SET BA
006760  012777  177760  171530          MOV     #-20,@FC        ;SET FC
006766  012777  177770  171516          MOV     #-10,@WC        ;SET WC
006774  012777  000013  171532          MOV     #13,@MR         ;SELECT WAM 2
007002  012777  000061  171500          MOV     #61,@C1         ;SET WRITE+GO
007010  052777  000020  171502          BIS     #20,@CS         ;FORCE EVEN PARITY
007016  012701  000040                  MOV     #40,R1          ;SET TEST WORD
007022  012703  000004                  MOV     #4,R3
007026  005000                          CLR     R0
007030  005300                  1$:     DEC     R0
007032  001376                          BNE     1$              ;DELAY
007034  005303                          DEC     R3
007036  001374                          BNE     1$
007040  012700  000004                  MOV     #4,R0
007044  012777  000013  171462  LT24B:  MOV     #13,@MR         ;CLOCK MR 4 TIMES
007052  005300                          DEC     R0
007054  022700  000002                  CMP     #2,R0           ;SEE IF DONE 1 BYTE
007060  001002                          BNE     LT24B0          ;IF NOT: BR
007062  017701  171446                  MOV     @MR,R1          ;ELSE GET BYTE 1
007066  005700                  LT24B0: TST     R0              ;SEE IF BYTE 2
007070  001365                          BNE     LT24B           ;IF NOT: BR
007072  017704  171436                  MOV     @MR,R4          ;GET BYTE 2
007076  005000                          CLR     R0
007100  005300                  LT24C:  DEC     R0
007102  001376                          BNE     LT24C           ;DELAY
007104  032777  000040  171412          BIT     #40,@ER         ;SEE IF DPAR IS SET
007112  001023                          BNE     LT24D           ;IF SO: BR
007114  000301                          SWAB    R1
007116  042701  177400                  BIC     #177400,R1      ;GET LOW BYTE
007122  042704  000377                  BIC     #377,R4
007126  050401                          BIS     R4,R1           ;GET HIGH BYTE
007130  005237  000740                  INC     T24FL           ;SET T24 FLAG
007134  012737  027712  000666          MOV     #TMS22,ERADD    ;SET ERROR CODE
007142  012737  000001  000712          MOV     #1,EXPT.FLG     ;SET EXPT FLG
007150  004737  015222                  JSR     PC,LTGER0       ;GO PRINT ERROR
007154  005037  000740                  CLR     T24FL           ;CLEAR FLAG
007160  000412                          BR      LT24X
007162  012701  000050          LT24D:  MOV     #50,R1
007166  017702  171332                  MOV     @ER,R2          ;GET ERROR REGISTER
007172  042702  020000                  BIC     #20000,R2       ;MASK OPI
007176  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
007200  001402                          BEQ     LT24X           ;IF NOT: BR
007202  004737  015210                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
007206  042777  000020  171304  LT24X:  BIC     #20,@CS         ;RESET EVEN PARITY
007214  004737  015054                  JSR     PC,ERRPA        ;GO DO FOR CLEAR
007220  004737  015670                  JSR     PC,DRVCLR       ;GO SEE IF DRIVE CLEAR OK
007224  004737  016600                  JSR     PC,ITER         ;GO SEE IF ITERATION
007230  012737  000020  000602          MOV     #20,ITAMI
007236  000137  002530                  JMP     TSCD2           ;RETURN TO SCHED
```

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

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

```
2485
2486                                        ;LOGIC TEST 27: ILLEGAL REGISTER(ILR)**********
2487
2488  007610  022737  172400  000510  LT27:   CMP     #172400,C1      ;SEE IF ADDRESSES OPEN
2489  007616  001041                          BNE     LT27XX          ;IF NOT: BR
2490  007620  012737  007644  000706          MOV     #LT27A,SCOLP    ;SET SCOPE ADDRESS
2491  007626  012737  025656  000616          MOV     #MSLT27,EMADDR  ;SET TEST HEADER
2492  007634  012700  000020          LT27IT: MOV     #20,R0          ;SET NUMBER OF ILR TESTS
2493  007640  012701  172434                  MOV     #172434,R1      ;SET FIRST ILR ADDRESS
2494  007644  004737  016754          LT27A:  JSR     PC,INITS        ;GO INIT, SELECT DRIVE+SLAVE
2495  007650  011103                          MOV     (R1),R3         ;ATTEMPT ILR READ
2496  007652  032777  000002  170644          BIT     #2,@ER          ;SEE IF ILR=1
2497  007660  001010                          BNE     LT27B           ;IF SO: BR
2498  007662  012737  000001  000712          MOV     #1,EXPT         ;SET EXPT-NOT RCVD FLAG
2499  007670  012737  027601  000666          MOV     #TMS10,ERADD    ;SET ERROR CODE
2500  007676  004737  015230                  JSR     PC,LTGER        ;GO PRINT ERROR
2501  007702  005300                  LT27B:  DEC     R0              ;SEE IF DONE ALL
2502  007704  001402                          BEQ     LT27X           ;IF SO: BR
2503  007706  005721                          TST     (R1)+           ;BUMP ADDRESS
2504  007710  000755                          BR      LT27A           ;CONTINUE TESTS
2505  007712  004737  016600          LT27X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2506  007716  004737  015670                  JSR     PC,DRVCLR
2507  007722  000137  002530          LT27XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2509
2510                                        ;LOGIC TEST 30: DRIVE TIMING ERROR**********
2511
2512  007726  012737  050006  000666  LT30:   MOV     #TMS32,ERADD    ;SET ERROR CODE
2513  007734  012737  025712  000616          MOV     #MSLT30,EMADDR  ;SET TEST HEADER
2514  007742  012737  007750  000706          MOV     #LT30IT,SCOLP   ;SET SCOPE ADDRESS
2515  007750  004737  016754          LT30IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE + SLAVE
2516  007754  052777  000300  170560          BIS     #300,@TC        ;SET NORMAL FORMAT
2517  007762  012701  010000                  MOV     #10000,R1       ;SET TEST WORD
2518  007766  012777  000017  170540          MOV     #17,@MR         ;CRIPPLE OCCUPIED
2519  007774  005077  170516                  CLR     @FC             ;SET FC3
2520  010000  012777  000061  170502          MOV     #61,@C1         ;LOAD WRITE+GO
2521  010006  032777  010000  170510          BIT     #10000,@ER      ;SEE IF DTE SET
2522  010014  001005                          BNE     LT30A           ;IF SO: BR
2523  010016  012737  000001  000712          MOV     #1,EXPL         ;SET EXPT FLG
2524  010024  004737  015222                  JSR     PC,LTGERO       ;GO PRINT ERROR
2525  010030  004737  016754          LT30A:  JSR     PC,INIT3        ;GO INIT SELECT DRIVE,SLAVE
2526  010034  052777  000300  170500          BIS     #300,@TC        ;SET FORMAT
2527  010042  012701  010000                  MOV     #10000,R1       ;SET TEST WORD
2528  010046  005077  170444                  CLR     @FC             ;SET FC3
2529  010052  012777  000015  170454          MOV     #15,@MR         ;SET WRAP 3
2530  010060  012777  000061  170422          MOV     #61,@C1         ;LOAD WRITE+GO
2531  010066  012704  040000                  MOV     #40000,R4
2532  010072  005777  170444          LT30B:  TST     @TC             ;SEE IF ALPHA
2533  010076  100015                          BPL     LT30C           ;AWAIT ALPHA
2534  010100  005300                          DEC     R0
2535  010102  001373                          BNE     LT30B
2536  010104  013704  000616                  MOV     EMADDR,R4
2537  010110  004737  017760                  JSR     PC,TTOUT        ;PRINT HEADER
2538  010114  012704  023125                  MOV     #MSG50,R4
2539  010120  004737  017760                  JSR     PC,TTOUT        ;PRINT ALPHA ERROR
2540  010124  004737  016550                  JSR     PC,SCOPE
2541  010130  000435                          BR      LT30X
2542  010132  012777  000015  170374  LT30C:  MOV     #15,@MR         ;CLOCK MR
2543  010140  012777  000015  170366          MOV     #15,@MR         ;CLOCK MR
2544  010146  005000                          CLR     R0
2545  010150  005300                  LT30D:  DEC     R0
2546  010152  001376                          BNE     LT30D           ;DELAY
2547  010154  032777  010000  170342          BIT     #10000,@ER      ;SEE IF DTE SET
2548  010162  001006                          BNE     LT30E           ;IF SO: BR
2549  010164  012737  000001  000712          MOV     #1,EXPL         ;SET EXPT FLG
2550  010172  004737  015222                  JSR     PC,LTGERO       ;GO PRINT ERROR
2551  010176  000412                          BR      LT30X
2552  010200  012701  010000          LT30E:  MOV     #10000,R1       ;SET TEST WORD
2553  010204  017702  170314                  MOV     @ER,R2          ;GET ERROR REGISTER
2554  010210  042702  020100                  BIC     #20100,R2       ;MASK OPI AND VPE
2555  010214  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2556  010216  001402                          BEQ     LT30X           ;IF NOT: BR
2557  010220  004737  015210                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2558  010224  004737  016600          LT30X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2559  010230  004737  015054                  JSR     PC,ERRPA        ;GO CLEAR GO BIT
2560  010234  004737  015670                  JSR     PC,DRVCLR
2561  010240  000137  002530                  JMP     @SCOPE          ;RETURN TO SCHED
2562
```

```
                                                       ;LOGIC TEST 31: OPERATION INCOMPLETE(OPI)**********

010244  012737  025750  000616  LT31:    MOV    #MSLT31,EMADDR   ;SET TEST HEADER
010252  012737  010252  000706  LT31IT:  MOV    #LT31IT,SCOLP    ;SET SCOPE ADDRESS
010260  012737  030022  000666           MOV    #TMS33A,ERADD    ;SET ERROR MSG HDR
010266  012737  000002  000602           MOV    #2,ITAMT         ;SET REDUCED ITER COUNT
010274  004737  016754                   JSR    PC,INIT3         ;INIT, SELECT DRIVE+SLAVE
010300  005000                           CLR    R0
010302  005300                  1$:      DEC    R0
010304  001376                           BNE    1$               ;AWAIT OPI RESET
010306  052777  000300  170226           BIS    #300,@TC         ;SET FORMAT
010314  012777  000013  170212           MOV    #13,@MR          ;SET WAM 2
010322  005077  170170                   CLR    @FC              ;SET FRAME COUNT
010326  012705  020000                   MOV    #20000,R5        ;SET TEST BIT (OPI)
010332  012702  010350                   MOV    #2$,R2           ;SET RETURN ADDRESS FROM TIMER
010336  004737  010550                   JSR    PC,TIMON         ;START TIMER
010342  012777  000061  170140           MOV    #61,@C1          ;LOAD WRITE+GO
010350  030577  170150          2$:      BIT    R5,@ER           ;BRANCH WHEN OPI SETS
010354  001002                           BNE    3$
010356  000163  010642                   JMP    TIMER(R3)        ;GO TO TIMER & RETURN TO 2$ ABOVE
010362  017702  170136          3$:      MOV    @ER,R2           ;GET ERROR REGISTER
010366  020502                           CMP    R5,R2            ;SEE IF UNEXPECTED ERRORS
010370  001403                           BEQ    4$               ;IF NOT; BR
010372  004737  015210                   JSR    PC,LTGER3        ;ELSE PRINT ERROR
010376  000453                           BR     LT31X
010400  004737  010734          4$:      JSR    PC,TIMOK         ;GO CHECK TIME FOR OPI TO SET
010404  102450                           BVS    LT31X            ;BRANCH IF TIME WAS INCORRECT

010406  012737  010422  000706           MOV    #LT31A,SCOLP     ;SET SCOPE LOOP
010414  012737  030036  000666           MOV    #TMS33B,ERADD    ;SET ERROR MSG HEADER
010422  004737  016754          LT31A:   JSR    PC,INIT3         ;GO INIT
010426  005000                           CLR    R0
010430  005300                  1$:      DEC    R0               ;WAIT FOR OPI TO CLEAR
010432  001376                           BNE    1$
010434  052777  000300  170100           BIS    #300,@TC         ;SET FORMAT
010442  012777  000015  170064           MOV    #15,@MR          ;SET WRAP 3
010450  012702  010472                   MOV    #2$,R2           ;SET RETURN ADDRESS FROM TIMER
010454  012705  020000                   MOV    #20000,R5        ;SET TEST WORD
010460  004737  010550                   JSR    PC,TIMON         ;START TIMER
010464  012777  000071  170016           MOV    #71,@C1          ;LOAD READ+GO
010472  030577  170026          2$:      BIT    R5,@ER           ;BRANCH WHEN OPI SETS
010476  001002                           BNE    3$
010500  000163  010642                   JMP    TIMER(R3)        ;GO TO TIMER
010504  017702  170014          3$:      MOV    @ER,R2           ;GET ERROR REGISTER
010510  020502                           CMP    R5,R2            ;SEE IF UNEXPECTED ERRORS
010512  001403                           BEQ    4$
010514  004737  015210                   JSR    PC,LTGER3        ;ELSE PRINT ERROR
010520  000402                           BR     LT31X            ;EXIT TEST
010522  004737  010734          4$:      JSR    PC,TIMOK         ;GO CHECK TIME
010526  004737  016600          LT31X:   JSR    PC,ITER          ;GO SEE IF ITERATIONS
010532  004737  015670                   JSR    PC,DRVCLR
010536  012737  000020  000602           MOV    #20,ITAMT
010544  000137  002530                   JMP    TSCD2            ;RETURN TO SCHED

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

```
                                        ;ENANCE REGISTER (BIT 6) THAT TOGGLES EVERY 56 (10) MICROSECONDS. THIS
                                        ;ROUTINE WAITS FOR THE OSCILLATOR TO TOGGLE AND RETURN WITH R3 INDICATING
                                        ;THE STATE OF THE OSCILLATOR.
010550  005000                          TIMON:  CLR     R0              ;CLEAR TICK COUNT
010552  005001                                  CLR     R1
010554  012703  000024                          MOV     #24,R3          ;PRESET INDEX TO TIMER
010560  032777  000100  167746                  BIT     #100,@MR        ;BRANCH IF OSC CLEAR
010566  001405                                  BEQ     2$
010570  032777  000100  167736  1$:             BIT     #100,@MR        ;WAIT FOR OSC TO CLEAR
010576  001374                                  BNE     1$
010600  000405                                  BR      4$              ;EXIT

010602  005403                          2$:     NEG     R3              ;SET INDEX TO TIMER
010604  032777  000100  167722  3$:             BIT     #100,@MR        ;WAIT FOR OSC TO SET
010612  001774                                  BEQ     3$
010614  000207                          4$:     RTS     PC              ;RETURN

                                        ;THIS ROUTINE TIMES AN EVENT. EACH TIME THE OSCILLATOR BIT CHANGES
                                        ;STATE THE TICK COUNT IN R1 & R0 IS INCREMENTED. THE ROUTINE IS CALLED
                                        ;USING R3 AS AN INDEX TO INDICATE THE OSCILLATORS PAST STATE. WHEN
                                        ;THE OSC BIT CHANGES STATE R3 IS NEGATED.
010616  032777  000100  167710  TIMER1: BIT     #100,@MR        ;BRANCH IF OSC HAS CHANGED STATE
010624  001406                                  BEQ     TIMER
010626  000112                                  JMP     (R2)            ;RETURN
        010642                                  .=TIMER1+24
010642  005403                          TIMER:  NEG     R3              ;SET INDEX TO OTHER STATE
010644  062700  000001                          ADD     #1,R0           ;INCREMENT TICK COUNT
010650  005501                                  ADC     R1
010652  022701  000003                          CMP     #3,R1           ;BRANCH IF TIMER OVERFLOWS
010656  001410                                  BEQ     TIMOVF
010660  000112                                  JMP     (R2)            ;RETURN
        010666                                  .=TIMER +24
010666  032777  000100  167640  TIMER0: BIT     #100,@MR        ;BRANCH IF OSC SET
010674  001362                                  BNE     TIMER
010676  000112                                  JMP     (R2)            ;RETURN

010700  013704  000616                  TIMOVF: MOV     EMADDR,R4       ;TYPE TEST HEADER
010704  004737  017760                          JSR     PC,TTOUT
010710  013704  000666                          MOV     ERADD,R4        ;GET ERROR MSG ADDRESS
010714  004737  017760                          JSR     PC,TTOUT        ;AND TYPE IT
010720  012704  030116                          MOV     #TMS33E,R4      ;TYPE
010724  004737  017760                          JSR     PC,TTOUT        ;'TIMER OVERFLOWED'
010730  000137  010526                          JMP     LT31X           ;GO EXIT TEST

                                        ;ROUTINE TO CHECK IF TIME IS WITHIN LIMITS. IF NOT THE ROUTINE RETURNS
                                        ;WITH THE 'V' BIT SET. THE LIMITS WERE SLECTED BY DIVIDING THE TIME
                                        ;IN MICROSECONDS BY 448. THE LOWER LIMIT IS 5,500.000 USECS (5.5 SECS);
                                        ;THE UPPER LIMIT IS 9,500,000 USECS (9.5 SECS). THE 448 IS DERIVED FROM
                                        ;56 USECS/TICK TIMES THE DIVISION BY 8 BY THE TIMOK ROUTINE.
010734  000240                          TIMOK:  NOP
010736  006201                                  ASR     R1              ;DIVIDE COUNT BY 8
010740  006000                                  ROR     R0
010742  006201                                  ASR     R1
010744  006000                                  ROR     R0
010746  006201                                  ASR     R1
010750  006000                                  ROR     R0
```

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

```
2710
2711
2712
2713                                        ;LOGIC TEST 32: UNSAFE(UNS)**********
2714
2715  011074  012737  026004  000616  LT32:   MOV     #MSLT32,EMADDR  ;SET TEST HEADER
2716  011102  012737  011110  000706          MOV     #LT32IT,SCOLP   ;SET SCOPE ADDRESS
2717  011110  004737  016754          LT32IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE +SLAVE
2718  011114  013700  000660                  MOV     SLVN,R0         ;GET SLAVE NUMBER
2719  011120  012701  040000                  MOV     #40000,R1       ;++B SET TEST WORD (UNS)
2720  011124  032777  000004  167404          BIT     #4,@DT          ;++B BRANCH IF TE16
2721  011132  001402                          BEQ     1$              ;++B
2722  011134  052701  004000                  BIS     #4000,R1        ;++B SET ALSO NEF FOR TU77
2723  011140  005100                  1$:     COM     R0              ;SET NONEXISTANT SLAVE
2724  011142  042700  177770                  BIC     #177770,R0      ;MASK SLAVE NUMBER
2725  011146  052700  000300                  BIS     #300,R0         ;SET FORMAT
2726  011152  010077  167364                  MOV     R0,@TC          ;SELECT ILLEGAL SLAVE
2727  011156  032777  002000  167352          BIT     #2000,@DT       ;EXIT TEST IF SALVE AVAILABLE
2728  011164  001030                          BNE     LT32XX
2729  011166  012777  000071  167314          MOV     #71,@C1         ;LOAD READ+GO
2730  011174  017702  167324                  MOV     @ER,R2          ;READ ER
2731  011200  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2732  011202  001011                          BNE     2$              ;IF SO: BR
2733  011204  012737  030136  000666          MOV     #TMS34,ERADD    ;SET ERROR CODE
2734  011212  012737  000001  000712          MOV     #1,EXFL         ;SET ERROR CODE
2735  011220  004737  015222                  JSR     PC,LTGER0       ;GO PRINT ERROR
2736  011224  000404                          BR      LT32X
2737  011226  020102                  2$:     CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2738  011230  001402                          BEQ     LT32X           ;IF NOT: BR
2739  011232  004737  015210                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2740  011236  004737  016600          LT32X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2741  011242  004737  015670                  JSR     PC,DRVCLR
2742  011246  000137  002530          LT32XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2744
2745                                        ;THE FOLLOWING 6 TESTS WILL LOOK AT VARIOUS BITS IN THE
2746                                        ;DRIVE STATUS(DS) AND TAPE CONTROL(TC)
2747                                        ;REGISTERS BY FORCING CERTAIN CONDITONS WHICH DO NOT
2748                                        ;REQUIRE TAPE MOVEMENT.
2749
2750                                        ;LOGIC TEST 33: POSITIONING IN PROGRESS(PIP)**********
2751
2752  011252  012737  026040  000616  LT33:   MOV     #MSLT33,EMADDR  ;SET TEST HEADER
2753  011260  012737  011266  000706          MOV     #LT33IT,SCOLP   ;SET SCOPE ADDRESS
2754  011266  004737  016754          LT33IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2755  011272  012777  000013  167234          MOV     #13,@MR         ;SET WAM 2
2756  011300  012777  177777  167210          MOV     #-1,@FC         ;SET FCS
2757  011306  012777  000031  167174          MOV     #31,@C1         ;LOAD SPACE FORWARD+GO
2758  011314  032777  020000  167200          BIT     #20000,@DS      ;SEE IF PIP=1
2759  011322  001010                          BNE     LT33X           ;IF SO: BR
2760  011324  012737  027631  000666          MOV     #TMS14,ERADD    ;SET ERROR CODE
2761  011332  012737  000001  000712          MOV     #1,EXFL         ;SET ERROR CODE
2762  011340  004737  015222                  JSR     PC,LTGERO       ;GO PRINT ERROR
2763  011344  004737  016600          LT33X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2764  011350  000137  002530                  JMP     TSCD2           ;RETURN TO SCHED
```

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

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

```
2789
2790                                        ;LOGIC TEST 35: SLAVE ADDRESS CHANGE (SAC)**********
2791
2792  011510  012737  030161  000666  LT35:   MOV     #IMS37,ERADD
2793  011516  012737  026130  000616          MOV     #MSLT35,EMADDR
2794  011524  004737  016754          LT35IT: JSR     PC,INIT3        ;INIT SELECT DRIVE, SLAVE
2795  011530  032777  000020  166764  1$:     BIT     #20,@DS         ;SEE IF SDWN IS RESET
2796  011536  001374                  BNE     1$              ;IF NOT: BR
2797  011540  052777  000300  166774          BIS     #300,@TC        ;SET FORMAT
2798  011546  012777  000015  166760          MOV     #15,@MR         ;SET WAM 3
2799  011554  012777  000071  166726          MOV     #71,@C1         ;LOAD READ+GO
2800  011562  032777  020000  166752          BIT     #20000,@TC      ;SEE IF SAC=0
2801  011570  001410                  BEQ     LT35A           ;IF SO: BR
2802  011572  012737  000002  000712          MOV     #2,EXFL         ;SET RCV-NOT EXPT FLAG
2803  011600  012737  011524  000706          MOV     #LT35IT,SCOLP   ;SET SCOPE ADDRESS
2804  011606  004737  015222                  JSR     PC,LTGERO       ;GO PRINT ERROR
2805  011612  004737  016754          LT35A:  JSR     PC,INIT3        ;INIT
2806  011616  005277  166720                  INC     @TC             ;BUMP SLAVE ADDRESS
2807  011622  032777  020000  166712          BIT     #20000,@TC      ;SEE IF SAC=1
2808  011630  001010                  BNE     LT35X           ;IF SO: BR
2809  011632  012737  011612  000706          MOV     #LT35A,SCOLP    ;SET SCOPE ADDRESS
2810  011640  012737  000001  000712          MOV     #1,EXFL         ;SE EXPT-NOT RCVD FLAG
2811  011646  004737  015222                  JSR     PC,LTGERO       ;GO PRINT ERROR
2812  011652  004737  016600          LT35X:  JSR     PC,ITER
2813  011656  000137  002530                  JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                          ;LOGIC TEST 36: FRAME COUNTER STATUS(FCS)**********

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

```
2836
2837                                        ;LOGIC TEST 37: ACCELERATION(ACCL)**********
2838
2839  012002  012737  026242  000616  LT37:   MOV     #MSLT3 ,EMADDR
2840  012010  012737  030175  000666          MOV     #TMS39,ERADD    ;SET ERROR CODE
2841  012016  004737  016754          LT37IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2842  012022  052777  000300  166512          BIS     #300,@TC        ;SET FORMAT
2843  012030  005777  166506                  TST     @TC             ;SEE IF ACCL=1
2844  012034  100410                          BMI     LT37A           ;IF SO: BR
2845  012036  012737  000001  000712          MOV     #1,EXFL
2846  012044  012737  012016  000706          MOV     #LT37IT,SCOLP   ;SET SCOPE ADDRESS
2847  012052  004737  015222                  JSR     PC,LTGERO       ;GO PRINT ERROR
2848  012056  004737  016754          LT37A:  JSR     PC,INIT3        ;INIT
2849  012062  052777  000300  166452          BIS     #300,@TC        ;SET FORMAT
2850  012070  012777  000015  166436          MOV     #15,@MR         ;SET WAM 3
2851  012076  012777  000071  166404          MOV     #71,@C1         ;LOAD READ+GO
2852  012104  012700  100000                  MOV     #100000,R0      ;SET ACCL DELAY
2853  012110  005777  166426          LT37B:  TST     @TC             ;SEE IF ACCL=0
2854  012114  100012                          BPL     LT37X           ;IF SO: BR
2855  012116  005300                          DEC     R0
2856  012120  001373                          BNE     LT37B           ;DELAY
2857  012122  012737  012056  000706          MOV     #LT37A,SCOLP    ;SET SCOPE ADDRESS
2858  012130  012737  000002  000712          MOV     #2,EXFL
2859  012136  004737  015222                  JSR     PC,LTGERO       ;GO PRINT ERROR
2860  012142  004737  016600          LT37X:  JSR     PC,ITER
2861  012146  000137  002530                  JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                ;LOGIC TEST 40: PE TAPE MARK (TM)**********

012152  012737  012166  000706  LT40:     MOV     #LT40IT,SCOLP      ;SET SCOPE ADDRESS
012160  012737  026310  000616            MOV     #MSLT40,EMADDR
012166  004737  016770          LT40IT:   JSR     PC,INIT4           ;INIT, SELECT DRIVE+SLAVE
012172  005000                            CLR     R0
012174  005300                  1$:       DEC     R0
012176  001376                            BNE     1$                 ;DELAY FOR OPI RESET
012200  052777  002300  166334            BIS     #2300,@TC
012206  012777  000007  166320            MOV     #7,@MR             ;SET WAM 0
012214  012777  000027  166266            MOV     #27,@C1            ;LOAD WRITE TAPE MARK+GO
012222  012700  100000                    MOV     #100000,R0         ;SET DELAY
012226  032777  000004  166266  2$:       BIT     #4,@DS             ;SEE IF TM=1
012234  001012                            BNE     LT40X              ;IF SO: BR
012236  005300                            DEC     R0
012240  001372                            BNE     2$                 ;DELAY
012242  012737  027527  000666            MOV     #TMS3,ERADD
012250  012737  000001  000712            MOV     #1,EXFL
012256  004737  015222                    JSR     PC,LTGERO          ;GO PRINT ERROR
012262  004737  016600          LT40X:    JSR     PC,ITER
012266  000137  002530          LT40XX:   JMP     TSCD2              ;RETURN TO SCHED
```

```
                                                ;LOGIC TEST 41: NRZ TAPE MARK (TM,VPE,ITM)**********
                                                
012272  012737  012306  000706  LT41:   MOV     #LT41IT,SCOLP   ;SET SCOPE ADDRESS
012300  012737  026355  000616          MOV     #MSLT41,EMADDR
012306  004737  016754          LT41IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE,SLAVE
012312  052777  001700  166222          BIS     #1700,@TC       ;SET NRZ+NORMAL FORMAT
012320  012777  177760  166170          MOV     #-20,@FC        ;SET FCS
012326  012777  000007  166200          MOV     #7,@MR          ;SET WAM 0
012334  012777  000027  166146          MOV     #27,@C1         ;LOAD WRITE TAPE MARK+GO
012342  005000                          CLR     R0
012344  032777  000004  166150  1$:     BIT     #4,@DS          ;SEE IF TM=1
012352  001012                          BNE     2$              ;IF SO: BR
012354  005300                          DEC     R0
012356  001372                          BNE     1$              ;DELAY
012360  012737  027527  000666          MOV     #TMS3,ERADD     ;SET ERROR CODE
012366  012737  000001  000712          MOV     #1,EXFL
012374  004737  015222                  JSR     PC,LTGER0       ;GO PRINT ERROR
012400  032777  002000  166116  2$:     BIT     #2000,@ER       ;SEE IF ITM=1
012406  001010                          BNE     3$              ;IF SO: BR
012410  012737  027772  000666          MOV     #TMS30,ERADD    ;SET ERROR CODE
012416  012737  000001  000712          MOV     #1,EXFL
012424  004737  015222                  JSR     PC,LTGER0       ;GO PRINT ERROR
012430  032777  000100  166066  3$:     BIT     #100,@ER        ;SEE IF VPE=1
012436  001011                          BNE     4$              ;IF SO: BR
012440  012737  027757  000666          MOV     #TMS28,ERADD    ;SET ERROR CODE
012446  012737  000001  000712          MOV     #1,EXFL
012454  004737  015222                  JSR     PC,LTGER0       ;GO PRINT ERROR
012460  000410                          BR      LT41X
012462  012701  002100          4$:     MOV     #2100,R1        ;SET EXPT ERROR BITS
012466  017702  166032                  MOV     @ER,R2          ;GET ERROR REGISTER
012472  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
012474  001402                          BEQ     LT41X           ;IF SO!: BR
012476  004737  015210                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
012502  005002          LT41X:  CLR     R2              ;SET TIMER
012504  032777  000200  166010  1$:     BIT     #200,@DS        ;SEE IF DRY SET
012512  001002                          BNE     2$              ;IF SO: BR
012514  005302                          DEC     R2              ;AWAIT DRY
012516  001372                          BNE     1$              ;DELAY
012520  004737  016600          2$:     JSR     PC,ITER         ;GO SEE IF ITERATIONS
012524  004737  015670                  JSR     PC,DRVCLR       ;GO DO DRIVE CLEAR
012530  000137  002530                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2929
2930                                    ;THE FOLLOWING SIX(6) TEST WILL REQUIRE TAPE MOVEMENT.  EACH
2931                                    ;TEST WILL PERFORM A TAPE WRITE WHILE IN A PARTICULAR MAINTENANCE
2932                                    ;MODE IN ORDER TO FORCE THE REMAINING ERROR CONDITIONS.
2933
2934                                    ;LOGIC TEST 42: CYCLIC REDUNDANCY ERROR(CRC)**********
2935
2936  012534  012737  001700  000772  LT42:   MOV     #1700,UDES      ;SET UNIT DESCRIPTION = NRZ
2937  012542  004737  015024          JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
2938  012546  012700  001000          MOV     #1000,R0
2939  012552  005300                  1$:     DEC     R0
2940  012554  001376                  BNE     1$              ;PAUSE
2941  012556  012737  026424  000616  MOV     #MSLT42,EMADDR
2942  012564  012737  012572  000706  MOV     #LT42IT,SCOLP   ;SET SCOPE ADDRESS
2943  012572  004737  017004          LT42IT: JSR     PC,INIT         ;INIT SELECT DRIVE+SLAVE
2944  012576  012777  177770  165706  MOV     #-10,@WC
2945  012604  012777  177760  165704  MOV     #-20,@FC        ;SET FC=20
2946  012612  012777  030204  165674  MOV     #WDATA,@BA      ;SET BUS ADDRESS
2947  012620  012777  000007  165706  MOV     #7,@MR          ;SET MM CODE
2948  012626  012777  000061  165654  MOV     #61,@C1         ;LOAD WRITE+GO
2949  012634  005000                  CLR     R0
2950  012636  032777  000200  165656  LT42A:  BIT     #200,@DS        ;SEE IF DRY=1
2951  012644  001002                  BNE     LT42B           ;IF SO: BR
2952  012646  005300                  DEC     R0
2953  012650  001372                  BNE     LT42A           ;DELAY
2954  012652  022777  000200  165644  LT42B:  CMP     #200,@ER        ;SEE IF LRC ERROR ONLY
2955  012660  001007                  BNE     LT42B1          ;IF NOT: BR
2956  012662  017702  165642          MOV     @CC,R2          ;GET CHECK CHAR
2957  012666  042702  177000          BIC     #177000,R2      ;MASK CRC
2958  012672  022702  000777          CMP     #777,R2         ;SEE IF SETUP CRC IS CORRECT
2959  012676  001410                  BEQ     LT42B2          ;IF SO: BR
2960  012700  004737  015210          LT42B1: JSR     PC,LTGER3       ;ELSE PRINT ERROR SETUP
2961  012704  012704  023216          MOV     #MSG55,R4
2962  012710  004737  017760          JSR     PC,TTOUT        ;PRINT SETUP ERROR MSG
2963  012714  000137  002530          JMP     TSCD2           ;RETURN TO SCHED
2964  012720  004737  017004          LT42B2: JSR     PC,INIT         ;GO INIT
2965  012724  012777  177770  165560  MOV     #-10,@WC        ;SET WC
2966  012732  012777  177760  165556  MOV     #-20,@FC        ;SET FC
2967  012740  012777  030204  165546  MOV     #WDATA,@BA      ;SET BA
2968  012746  012777  000021  165560  MOV     #21,@MR         ;SET MM
2969  012754  012777  000061  165526  MOV     #61,@C1         ;LOAD WRITE+GO
2970  012762  005000                  CLR     R0
2971  012764  032777  000200  165530  LT42C:  BIT     #200,@DS        ;SEE IF DRY
2972  012772  001002                  BNE     LT42D           ;IF SO: BR
2973  012774  005300                  DEC     R0
2974  012776  001372                  BNE     LT42C           ;AWAIT DRY
2975  013000  005777  165520          LT42D:  TST     @ER             ;SEE IF CRC=1
2976  013004  100411                  BMI     LT42E           ;IF SO: BR
2977  013006  012737  030153  000666  MOV     #IM536,ERADD    ;SET ERROR CODE
2978  013014  012737  000001  000712  MOV     #1,EXPT
2979  013022  004737  015222          JSR     PC,LTGER0       ;GO PRINT ERROR
2980  013026  000410                  BR      LT42X
2981  013030  012701  100200          LT42E:  MOV     #100200,R1      ;SET EXPT ERROR BITS
2982  013034  017702  165464          MOV     @ER,R2          ;GET ERROR REGISTER
2983  013040  020102                  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2984  013042  001402                  BEQ     LT42X           ;IF NOT: BR
```

```
 2985  013044  004737  015210                JSR     PC,LTGER3       ;ELSE PRINT ERROR
 2986  013050  004737  016600        LT43:   JSR     PC,ITER         ;DO ITERATIONS
 2987  013054  004737  015670                JSR     PC,DRVCLR
 2988  013060  000137  002530                JMP     TSCD2           ;RETURN TO SCHED
 2989
 2990                                        ;LOGIC TEST 43: LONGITUALINAL REDUNDANCY(LRC)**********
 2991
 2992  013064  032777  000004  165444  LT43:   BIT     #4,@DT          ;**B BRANCH IF NOT A TE16
 2993  013072  001114                    BNE     LT43XX          ;**B
 2994  013074  012737  001700  000772          MOV     #1700,UDES      ;SET UNIT DESCRIPTION = NRZ
 2995  013102  004737  015024                JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
 2996  013106  012737  013122  000706          MOV     #LT43IT,SCOLP   ;SET SCOPE ADDRESS
 2997  013114  012737  026460  000616          MOV     #MSLT43,EMADDR
 2998  013122  004737  017004        LT43IT: JSR     PC,INIT         ;INIT, SELECT DRIVE+SLAVE
 2999  013126  005001                    CLR     R1
 3000  013130  005301                1$:     DEC     R1              ;DELAY
 3001  013132  001376                    BNE     1$
 3002  013134  012777  000023  165372          MOV     #23,@MR         ;SET MM
 3003  013142  012777  177770  165342          MOV     #-10,@WC        ;SET WC
 3004  013150  012777  177760  165340          MOV     #-20,@FC        ;SET FC
 3005  013156  012777  030200  165330          MOV     #WDATA,@BA      ;SET BA
 3006  013164  012777  000061  165316          MOV     #61,@C1         ;LOAD WRITE+GO
 3007  013172  005000                    CLR     R0
 3008  013174  032777  000200  165320  LT43C:  BIT     #200,@DS        ;SEE IF DRY
 3009  013202  001002                    BNE     LT43D           ;IF SO: BR
 3010  013204  005300                    DEC     R0
 3011  013206  001372                    BNE     LT43C           ;AWAIT DRY
 3012  013210  032777  000200  165306  LT43D:  BIT     #200,@ER        ;SEE IF LRC=1
 3013  013216  001011                    BNE     LT43E           ;IF SO: BR
 3014  013220  012737  027743  000666          MOV     #TMS26,ERADD    ;SET ERROR CODE
 3015  013226  012737  000001  000712          MOV     #1,EXPT
 3016  013234  004737  015222                JSR     PC,LTGER0       ;GO PRINT
 3017  013240  000425                    BR      LT43X
 3018  013242  017702  165266          LT43F:  MOV     @MR,R2
 3019  013246  042702  000177                BIC     #177,R2         ;MASK LRC
 3020  013252  012701  157600                MOV     #157600,R1      ;SET EXPT LRC
 3021  013256  020102                    CMP     R1,R2           ;SEE IF EXPT = RCVD
 3022  013260  001405                    BEQ     LT43F           ;IF SO: BR
 3023  013262  012737  023173  000666          MOV     #MSG53,ERADD    ;SET ERROR CODE
 3024  013270  004737  016336                JSR     PC,LTGER1       ;PRINT ERROR
 3025  013274  017702  165224          LT43F:  MOV     @ER,R2          ;GET ERROR REGISTER
 3026  013300  012701  000200                MOV     #200,R1         ;SET EXPT ERROR BITS
 3027  013304  020102                    CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
 3028  013306  001402                    BEQ     LT43X           ;IF NOT: BR
 3029  013310  004737  015210                JSR     PC,LTGER3       ;ELSE PRINT ERROR
 3030  013314  004737  016600        LT43X:  JSR     PC,ITER
 3031  013320  004737  015670                JSR     PC,DRVCLR
 3032  013324  000137  002530        LT43XX: JMP     TSCD2           ;RETURN TO SCHED
```

C 7


```
                                                       ;LOGIC TEST 44: PE CORRECTABLE DATA (CORR)***********

013330  012737  002300  000772  LT44:    MOV     #2300,UDES         ;SET UNIT DESRIPTION = PE
013336  004737  015024                   JSR     PC,STATIC          ;GO SEE IF STATIC ONLY
013342  012737  026514  000616           MOV     #MSLT44,EMADDR     ;SET HEADER
013350  012737  013356  000706           MOV     #LT44IT,SCOLP      ;SET SCOP
013356  004737  017004          LT44IT:  JSR     PC,INIT            ;GO INITIALIZE
013362  012777  177600  165122           MOV     #-200,@WC          ;SET WC=200
013370  012777  177400  165120           MOV     #-400,@FC          ;SET FC=400
013376  012777  030204  165110           MOV     #WDATA,@BA         ;SET BA=START OF WRITE BUFFER
013404  012777  000061  165076           MOV     #61,@C1            ;LOAD WRITE AND GO
013412  005000                           CLR     R0
013414  005777  165076          LT44A:   TST     @FC                ;SEE IF FC=0
013420  001402                           BEQ     LT44A1             ;IF SO;BR
013422  005300                           DEC     R0
013424  001373                           BNE     LT44A              ;AWAIT FC=0
013426  012777  000021  165100  LT44A1:  MOV     #21,@MR            ;SET MAINT MODE
013434  005000                           CLR     R0
013436  032777  000200  165056  LT44B:   BIT     #200,@DS           ;SEE IF DRY
013444  001002                           BNE     LT44C              ;IF SO ;BR
013446  005300                           DEC     R0
013450  001372                           BNE     LT44B              ;AWAIT DRY
013452  005777  165046          LT44C:   TST     @ER                ;SEE IF CORR=1
013456  100410                           BMI     LT44D              ;IF SO; BR
013460  012737  030144  000666           MOV     #TMS35,ERADD       ;ELSE SET ERROR CODE
013466  012737  000001  000712           MOV     #1,EXFL            ;SET EXPT FLAG
013474  004737  015222                   JSR     PC,LTGER0          ;GO PRINT ERROR
013500  000240                  LT44D:   NOP
013502  122777  000002  165020  LT44E:   CMPB    #2,@CC             ;SEE IF DEAD TRACK BIT 1
013510  001414                           BEQ     LT44F              ;IF SO; BR
013512  117702  165012                   MOVB    @CC,R2             ;ELSE SAVE RECVD
013516  042702  177000                   BIC     #177000,R2         ;MASK OUT CRC
013522  112701  000002                   MOVB    #2,R1              ;SAVE EXPT
013526  012737  022602  000666           MOV     #MSG42,ERADD       ;SET ERROR CODE
013534  004737  016336                   JSR     PC,LTGER1          ;GO PRINT ERROR
013540  000410                           BR      LT44X
013542  017702  164756          LT44F:   MOV     @ER,R2             ;GET ERROR REGISTER
013546  012701  100000                   MOV     #100000,R1         ;SET EXPT ERROR BITS
013552  020102                           CMP     R1,R2              ;SEE IF EXPT=RCVD
013554  001402                           BEQ     LT44X              ;IF SO; BR
013556  004737  015210                   JSR     PC,LTGER3          ;ELSE PRINT ERROR
013562  004737  016600          LT44X:   JSR     PC,ITER            ;GO SEE IF ITERATIONS
013566  004737  015620                   JSR     PC,DRVCLR          ;GO DO DRIVE CLEAR
013572  000137  002530          LT44XX:  JMP     TSCD2              ;RETURN TO SCHED
```

```

                                                        ;LOGIC TEST 45: PE INCORRECTABLE DATA(INC)***********

013576  012737  002300  000772  LT45:    MOV     #2300,UDES          ;SET UNIT DESCRIPTION = PE
013604  004737  015024                   JSR     PC,STATIC           ;GO SEE IF STATIC ONLY
013610  012737  026574  000616           MOV     #MSLT45,EMADDR
013616  012737  013624  000706           MOV     #LT45IT,SCOLP
013624  004737  017004          LT45IT:  JSR     PC,INIT             ;INIT SELECT DRIVE SLAVE
013630  012777  177600  164654           MOV     #-200,@WC           ;SET WC=200
013636  012777  177400  164652           MOV     #-400,@FC           ;SET FC=400
013644  012777  030204  164642           MOV     #WDATA,@BA          ;SET BA=START OF WRITE BUFFER
013652  012777  000061  164630           MOV     #61,@C1             ;LOAD WRITE+GO
013660  005000                           CLR     R0
013662  005777  164630          LT45E:   TST     @FC                 ;AWAIT FC=0
013666  001402                           BEQ     LT45E1
013670  005300                           DEC     R0
013672  001373                           BNE     LT45E               ;AWAIT FC=0
013674  012777  000023  164632  LT45E1:  MOV     #23,@MR             ;SET MAINT CODE
013702  005000                           CLR     R0
013704  032777  000200  164610  LT45A:   BIT     #200,@DS            ;SEE IF DRY IS SET
013712  001002                           BNE     LT45B               ;IF SO: BR
013714  005300                           DEC     R0
013716  001372                           BNE     LT45A               ;AWAIT DRY
013720  032777  000100  164576  LT45B:   BIT     #100,@ER            ;SEE IF INC=1
013726  001010                           BNE     LT45D               ;IF SO:BR
013730  012737  027721  000666           MOV     #TMS23,ERADD        ;SET ERROR CODE
013736  012737  000001  000712           MOV     #1,EXFL
013744  004737  015222                   JSR     PC,LTGER0           ;GO PRINT ERROR
013750  017702  164554          LT45D:   MOV     @CC,R2              ;GET CHECK CHAR
013754  042702  177000                   BIC     #177000,R2          ;MASK CHECK CHAR
013760  012701  000046                   MOV     #46,R1              ;SET EXPT CK
013764  020102                           CMP     R1,R2               ;SEE IF EXPT = RCVD
013766  001405                           BEQ     LT45F               ;IF SO: BR
013770  012737  023205  000666           MOV     #MSG54,ERADD
013776  004737  016336                   JSR     PC,LTGER1           ;ELSE GO PRINT ERROR
014002  017702  164516          LT45F:   MOV     @ER,R2
014006  042702  120600                   BIC     #120600,R2          ;MASK OPI ,NSG, CORR, AND PEF
014012  012701  000100                   MOV     #100,R1             ;SET EXPT ERROR BITS
014016  020102                           CMP     R1,R2               ;SEE IF UNEXPECTED ERRORS
014020  001402                           BEQ     LT45X               ;IF NOT: BR
014022  004737  015210                   JSR     PC,LTGER3           ;ELSE PRINT ERROR
014026  004737  016600          LT45X:   JSR     PC,ITER
014032  004737  015670                   JSR     PC,DRVCLR
014036  000137  002530          LT45XX:  JMP     TSCD2               ;RETURN TO SCHED
```

```
                                                  ;LOGIC TEST 46: PE FORMAT ERROR(PEF,NSG)***********

014042  012737  002300  000772  LT46:    MOV     #2300,UDES         ;SET UNIT DESCRIPTION = PE
014050  004737  015024                   JSR     PC,STATIC          ;GO SEE IF STATIC ONLY
014054  012737  026656  000616           MOV     #MSLT46,EMADDR     ;SET HEADER
014062  012737  014070  000706           MOV     #LT46IT,SCOLP      ;SET SCOPE ADDRESS
014070  004737  017004          LT46IT:  JSR     PC,INIT            ;INITIALIZE
014074  012777  177770  164410           MOV     #-10,@WC           ;SET WC=10
014102  012777  177760  164406           MOV     #-20,@FC           ;SET FC=20
014110  012777  030204  164376           MOV     #WDATA,@BA         ;SET BA=START OF WRITE BUFFER
014116  012777  000061  164364           MOV     #61,@C1            ;LOAD WRITE+GO
014124  005777  164366          LT46A:   TST     @FC
014130  001375                           BNE     LT46A              ;AWAIT FC=0
014132  032777  000100  164374  1$:      BIT     #100,@MR           ;WAIT FOR TAPE TO START WRITEING POSTAMBLE
014140  001774                           BEQ     1$                 ;DELAY
014142  032777  000100  164364  2$:      BIT     #100,@MR
014150  001374                           BNE     2$
014152  012777  000027  164354           MOV     #27,@MR            ;SET MM CODE TO KILL PEF
014160  005000                           CLR     R0                 ;INIT TIMING LOOP
014162  032777  000200  164332  LT46B:   BIT     #200,@DS           ;SEE IF DRY SET
014170  001002                           BNE     LT46C              ;IF SO: BR
014172  005300                           DEC     R0
014174  001372                           BNE     LT46B              ;AWAIT DRY
014176  032777  000200  164320  LT46C:   BIT     #200,@ER           ;SEE IF PEF SET
014204  001011                           BNE     LT46D              ;IF SO: BR
014206  012737  027735  000666           MOV     #TMS25,ERADD       ;SET ERROR TAG
014214  012737  000001  000712           MOV     #1,EXFL            ;SET EXPT FLAG
014222  004737  015222                   JSR     PC,LTGER0          ;GO PRINT ERROR
014226  000420                           BR      LT46X
014230  017702  164270          LT46D:   MOV     @ER,R2             ;GET ERROR REGISTER
014234  042702  120100                   BIC     #120100,R2         ;++B CLEAR CRC,OPI & INC BITS (MAY OR MAY NOT SET)
014240  012701  000600                   MOV     #600,R1            ;++B SET EXPT ERROR BITS (NSG + PEF)
014244  032777  000004  164264           BIT     #4,@DT             ;++B BRANCH IF TE16
014252  001402                           BEQ     1$                 ;++B
014254  042701  000400                   BIC     #400,R1            ;++B TU77 SHOULD NOT SET NSG BIT
014260  020102                  1$:      CMP     R1,R2              ;SEE IF UNEXPECTED ERRORS
014262  001402                           BEQ     LT46X              ;IF NOT: BR
014264  004737  015210                   JSR     PC,LTGER3          ;ELSE PRINT ERROR
014270  004737  016600          LT46X:   JSR     PC,ITER
014274  004737  015670                   JSR     PC,DRVCLR
014300  000137  002530          LT46XX:  JMP     TSCD2              ;RETURN TO SCHED
```

```
                                                      ;LOGIC TEST 47: FRAME COUNT OVERFLOW(M8905-YB)*******

014304  012737  026712  000616  LT47:    MOV    #MSLT47,EMADDR   ;SE. TEST HEADER
014312  012737  014320  000706           MOV    #LT47IT,SCOLP    ;SET SCOPE ADDRESS
014320  004737  016754          LT47IT:  JSR    PC,INIT3         ;GO INIT
014324  012777  177770  164160           MOV    #-10,@WC         ;SET WC = 10
014332  012777  177760  164156           MOV    #-20,@FC         ;SET FC = 20
014340  052777  001700  164174           BIS    #1700,@TC        ;SET TO NRZ, NORMAL, ODD
014346  012777  030204  164140           MOV    #WDATA,@BA       ;SET BUS ADDRESS
014354  012777  000013  164152           MOV    #13,@MR          ;SET WRAP 2
014362  012777  000061  164120           MOV    #61,@C1          ;LOAD WRITE+GO
014370  012700  040000                   MOV    #40000,R0
014374  005777  164142          LT47A:   TST    @TC              ;SEE IF ALPHA
014400  100002                           BPL    LT47B            ;IF SO: BR
014402  005300                           DEC    R0
014404  001373                           BNE    LT47A            ;AWAIT ALPHA
014406  012700  000020          LT47B:   MOV    #20,R0           ;SET CLK CNT
014412  052777  000040  164114  LT47C:   BIS    #40,@MR
014420  042777  000040  164106           BIC    #40,@MR          ;CLOCK MR
014426  005300                           DEC    R0
014430  001370                           BNE    LT47C            ;IF NOT DONE ALL: BR
014432  017702  164060                   MOV    @FC,R2
014436  005001                           CLR    R1               ;SET TEST WORD
014440  020102                           CMP    R1,R2            ;SEE IF EXPT = RCVD
014442  001410                           BEQ    LT47X            ;IF SO: BR
014444  012737  022055  000666           MOV    #MSG19,ERADD     ;SET ERROR CODE
014452  012737  000001  000712           MOV    #1,EXFL          ;SET EXPT FLAG
014460  004737  016336                   JSR    PC,LTGER1        ;GO PRINT ERROR
014464  004737  016600          LT47X:   JSR    PC,ITER          ;GO SEE IF ITERATIONS
014470  000137  002530                   JMP    TSCD2            ;RETURN TO SCHEDULAR

```

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

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

014650  012737  027042  000616  LT51:   MOV     #MSLT51,EMADDR   ;;SET ERROR POINTER FOR STANDARD CONF.
014656  012737  014664  000706          MOV     #LT51IT,SCOLP    ;SET SCOPE LOOP ADDRS.
014664  004737  016770          LT51IT: JSR     PC,INIT4         ;SET SLAVE = PE
014670  042777  002300  163644          BIC     #2300,@TC        ;CLEAR DENSITY BITS
014676  052777  001300  163636          BIS     #1300,@TC        ;SET DENSITY = NRZ
014704  012777  177770  163600          MOV     #-10,@WC         ;SET WORD COUNT
014712  012777  177760  163576          MOV     #-20,@FC         ;SET FRAME COUNT
014720  012777  030204  163566          MOV     #WDATA,@BA       ;SET BUS ADDRESS
014726  012777  000013  163600          MOV     #13,@MR          ;SET WRAP 2
014734  012777  000061  163546          MOV     #61,@C1          ;SET WRITE COMMAND AND GO
014742  000240                          NOP
014744  000240                          NOP
014746  000240                          NOP
014750  012701  004000                  MOV     #4000,R1         ;SET EXPECTED RESULT
014754  017702  163544                  MOV     @ER,R2           ;GET ERROR REGISTER
014760  030102                          BIT     R1,R2            ;BRANCH IF NEF SET
014762  001006                          BNE     1$
014764  012737  000001  000712          MOV     #1,EXFL          ;SET EXPECTED FLAG
014772  004737  015222                  JSR     PC,LTGER0        ;PRINT ERROR MSG
014776  000404                          BR      LT51X
015000  020102                  1$:     CMP     R1,R2            ;BRANCH IF NO UNEXPECTED
015002  001402                          BEQ     LT51X            ;ERROR BITS WERE SET
015004  004737  015210                  JSR     PC,LTGER3
015010  004737  016600          LT51X:  JSR     PC,ITER          ;;ITERATE TEST
015014  004737  015670                  JSR     PC,DRVCLR        ;CLEAR DRIVE
015020  000137  002530                  JMP     TSCD2            ;RETURN TO SCHEDULER
```

```
3261                                        ;STATIC TESTS ONLY SUBROUTINE*********
3262
3263  015024  005737  000734        STATIC: TST     STFLG           ;SEE IF SINGLE TEST ONLY
3264  015030  001006                        BNE     1$              ;IF SO: BR
3265  015032  005737  001006                TST     STATC           ;SEE IF STATIC ONLY
3266  015036  001403                        BEQ     1$              ;IF NOT: BR
3267  015040  005726                        TST     (SP)+           ;RESET STACK
3268  015042  000137  002530                JMP     TSCD2           ;RETURN TO SCHEDULAR
3269  015046  005037  001002          1$:   CLR     RDRVF
3270  015052  000207                        RTS     PC              ;RETURN TO TEST
3271
```

```
3273
3274                                        ;END OF RECORD FORCE SUBROUTINE*****
3275
3276  015054  017700  163454        EORPA:  MOV     @MR,R0          ;GET MAINT REG
3277  015060  042700  000036                BIC     #36,R0          ;CLEAR CURRENT OP CODE
3278  015064  052700  000024                BIS     #24,R0          ;SET EOR CLEAR OP CODE
3279  015070  010077  163440                MOV     R0,@MR          ;DO EOR
3280  015074  042777  000037  163432        BIC     #37,@MR         ;CLEAR EOR AND MM
3281  015102  005000                        CLR     R0
3282  015104  012701  000002                MOV     #2,R1
3283  015110  032777  000001  163372 EORP1:  BIT     #1,@C1          ;SEE IF GO GONE
3284  015116  001430                        BEQ     EORP2           ;IF SO: BR
3285  015120  005300                        DEC     R0
3286  015122  001372                        BNE     EORP1           ;AWAIT GO RESET
3287  015124  005301                        DEC     R1
3288  015126  001370                        BNE     EORP1
3289  015130  032777  020000  163432        BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
3290  015136  001020                        BNE     EORP2           ;IF SO: BR
3291  015140  005737  000614                TST     HDRFL           ;SEE IF DONE HEADER
3292  015144  001004                        BNE     EORP1A          ;IF SO: BR
3293  015146  013704  000616                MOV     EMADDR,R4
3294  015152  004737  017760                JSR     PC,TTOUT        ;PRINT HEADER
3295  015156  012704  030617        EORP1A: MOV     #WMSG31,R4
3296  015162  004737  017760                JSR     PC,TTOUT        ;PRINT EOR GO BIT ERROR
3297  015166  032777  100000  163374        BIT     #100000,@SWR    ;SEE IF HALT ON ERROR
3298  015174  001401                        BEQ     EORP2           ;IF NOT: BR
3299  015176  000000                        HALT
3300  015200  000240        EORP2:  NOP
3301  015202  005037  000672 EORPX:  CLR     TEMP2           ;CLEAR FLAG
3302  015206  000207                        RTS     PC              ;RETURN
3303
```

```
3305                                            ;LOGIC TEST ADDRESSING ERROR SUBROUTINE**********
3306
3307  015210  005037  000712          LTGER3: CLR     EXFL
3308  015214  012737  023144  000666          MOV     #MSG51,ERADD
3309  015222  012737  000001  000742  LTGER0: MOV     #1,ADDFL        ;SET NO ADDRESS FLAG
3310  015230  000240                  LTGER:  NOP
3311  015232  005037  000662                  CLR     PFLG            ;CLEAR PRINT FLAG
3312  015236  032777  020000  163324          BIT     #20000,@SWR     ;SEE IF SHOULD PRINT
3313  015244  001112                          BNE     LTGX            ;IF NOT: BR
3314  015246  005737  000614          LTGA:   TST     HDRFL           ;SEE IF PRINTED HEADER
3315  015252  001004                          BNE     LTGA1           ;IF SO: BR
3316  015254  013704  000616                  MOV     EMADDR,R4
3317  015260  004737  017760                  JSR     PC,TTOUT        ;PRINT TEST HEADER
3318  015264  012737  000001  000614  LTGA1: MOV     #1,HDRFL        ;SET HEADER FLAG
3319  015272  013704  000666                  MOV     ERADD,R4
3320  015276  004737  017760                  JSR     PC,TTOUT        ;PRINT CONDITION ERROR
3321  015302  005737  000742                  TST     ADDFL
3322  015306  001003                          BNE     LTGA2
3323  015310  010103                          MOV     R1,R3
3324  015312  004737  020172                  JSR     PC,OCTP         ;PRINT ADDRESS
3325  015316  005737  000712          LTGA2:  TST     EXFL
3326  015322  001412                          BEQ     LTGC            ;IF NO STATUS: BR
3327  015324  012704  021435                  MOV     #MSG6,R4
3328  015330  022737  000001  000712          CMP     #1,EXFL         ;EXPT NOT RCVD
3329  015336  001402                          BEQ     LTGB
3330  015340  012704  021454                  MOV     #MSG7,R4        ;RCVD NOT EXPT
3331  015344  004737  017760          LTGB:   JSR     PC,TTOUT        ;PRINT STATUS
3332  015350  005237  000662          LTGC:   INC     PFLG
3333  015354  005737  000742                  TST     ADDFL           ;SEE IF ADD TST
3334  015360  001430                          BEQ     LTGD            ;IF SO: BR
3335  015362  005737  000740                  TST     T24FL           ;SEE IF TEST 24
3336  015366  001423                          BEQ     LTGC0           ;IF NOT: BR
3337  015370  012704  030604                  MOV     #WMSG27,R4
3338  015374  004737  017760                  JSR     PC,TTOUT        ;PRINT DATA TAG
3339  015400  012704  021755                  MOV     #MSG12,R4
3340  015404  004737  017760                  JSR     PC,TTOUT        ;PRINT EXPT TAG
3341  015410  012703  177777                  MOV     #-1,R3
3342  015414  004737  020162                  JSR     PC,OCTPE        ;PRINT EXPT
3343  015420  012704  021764                  MOV     #MSG13,R4
3344  015424  004737  017760                  JSR     PC,TTOUT        ;PRINT RCVD TAG
3345  015430  010103                          MOV     R1,R3           ;GET RCVD
3346  015432  004737  020162                  JSR     PC,OCTPE        ;PRINT RCVD
3347  015436  004737  015534          LTGC0:  JSR     PC,REGP         ;PRINT REGISTERS
3348  015442  032777  004000  163120  LTGD:   BIT     #4000,@SWR
3349  015450  001010                          BNE     LTGX
3350  015452  012704  022026                  MOV     #MSG16,R4
3351  015456  004737  017760                  JSR     PC,TTOUT
3352  015462  013703  000676                  MOV     ITCNT,R3        ;PRINT ITERATION
3353  015466  004737  020172                  JSR     PC,OCTP
3354  015472  005777  163072          LTGX:   TST     @SWR
3355  015476  100001                          BPL     LTGXA           ;IF NOT STOP ON ERROR: BR
3356  015500  000000                          HALT
3357  015502  005737  000662          LTGXA:  TST     PFLG
3358  015506  001004                          BNE     LTGXX           ;IF PRINTED: BR
3359  015510  032777  020000  163052          BIT     #20000,@SWR
3360  015516  001653                          BEQ     LTGA
```

```
3361  015520  005037  000742          LTGXX:  CLR     ADDFL           ;CLEAR ADDRESS FLAG
3362  015524  005037  000712                  CLR     EXFL
3363  015530  000137  016550                  JMP     SCOPE
3364
3365                                          ;SUBROUTINE TO PRINT MAJOR REGISTERS**********
3366
3367  015534  000240                  REGP:   NOP
3368  015536  012704  022767                  MOV     #MSG46,R4
3369  015542  004737  017760                  JSR     PC,TTOUT        ;PRINT REGISTER HEADER
3370  015546  017703  162736                  MOV     @C1,R3
3371  015552  004737  020162                  JSR     PC,OCTPE
3372  015556  017703  162730                  MOV     @WC,R3
3373  015562  004737  020162                  JSR     PC,OCTPE
3374  015566  017703  162722                  MOV     @BA,R3
3375  015572  004737  020162                  JSR     PC,OCTPE
3376  015576  017703  162714                  MOV     @FC,R3
3377  015602  004737  020162                  JSR     PC,OCTPE
3378  015606  017703  162706                  MOV     @CS,R3
3379  015612  004737  020162                  JSR     PC,OCTPE
3380  015616  017703  162700                  MOV     @DS,R3
3381  015622  004737  020162                  JSR     PC,OCTPE                ;PRINT REGISTERS
3382  015626  017703  162672                  MOV     @ER,R3
3383  015632  004737  020162                  JSR     PC,OCTPE
3384  015636  017703  162664                  MOV     @AS,R3
3385  015642  004737  020162                  JSR     PC,OCTPE
3386  015646  017703  162660                  MOV     @MR,R3
3387  015652  004737  020162                  JSR     PC,OCTPE
3388  015656  017703  162660                  MOV     @IC,R3
3389  015662  004737  020162                  JSR     PC,OCTPE
3390  015666  000207                  RTS     PC
3391
3392
```

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

MSG68	023730	1915	4030#										
MSG69	024002	1911	4031#										
MSG7	021454	3330	3971#										
MSG8	021472	1878	3972#										
MSG8A	021533	1876	3973#										
MSG9	021662	1932	3975#										
MSLT1	024012	1745	4034#										
MSLT10	024501	2033	4041#										
MSLT11	024557	2060	4042#										
MSLT12	024624	2101	4043#										
MSLT13	024675	2133	4044#										
MSLT14	024742	2160	4045#										
MSLT15	025051	2183	4046#										
MSLT16	025141	2205	4047#										
MSLT17	025230	2231	4048#										
MSLT2	024066	1798	4035#										
MSLT20	025311	2256	4049#										
MSLT21	025370	2285	4050#										
MSLT22	025424	2315	4051#										
MSLT23	025461	2339	4052#										
MSLT24	025526	2364	4053#										
MSLT25	025566	2422	4054#										
MSLT26	025622	2447	4055#										
MSLT27	025656	2491	4056#										
MSLT3	024145	1834	4036#										
MSLT30	025712	2513	4057#										
MSLT31	025750	2567	4058#										
MSLT32	026004	2715	4059#										
MSLT33	026040	2752	4060#										
MSLT34	026074	2770	4061#										
MSLT35	026130	2793	4062#										
MSLT36	026175	2818	4063#										
MSLT37	026242	2839	4064#										
MSLT4	024232	1931	4037#										
MSLT40	026310	2867	4065#										
MSLT41	026355	2890	4066#										
MSLT42	026424	2941	4067#										
MSLT43	026460	2997	4068#										
MSLT44	026514	3038	4069#										
MSLT45	026574	3084	4070#										
MSLT46	026656	3129	4071#										
MSLT47	025712	3169	4072#										
MSLT5	024314	1941	4038#										
MSLT50	026762	3202	4073#										
MSLT51	027042	3234	4074#										
MSLT6	024363	1981	4039#										
MSLT7	024432	2008	4040#										
MSS7A	023356	1627	4019#										
MTINT	017246	1334	1613	3654#									
NONSTD	001020	1451#											
NOST	016670	1636	3570#										
NRZOF	000656	1402#											
NXTDRV	002336	1700#											
NXTSLV	002410	1700#	1726										
OCTP	020172	1597	1606	1640	1649	1658	1668	1730	3304	3353	3507	3511	3517	3830#
OCTPE	020162	3342	3346	3371	3373	3375	3377	3379	3381	3383	3385	3387	3389	3828#

```
3488                                        ;LOGIC TEST REGISTER BIT ERROR SUBROUTIINE**********
3489
3490  016330  012737  000001  000732  LTGER0: MOV     #1,PEXFL        ;SET FLAG
3491  016336  000240          000000  LTGER1: NOP
3492  016340  005037  000662          CLR     PFLG            ;CLEAR PRINT FLAG
3493  016344  032777  020000  162216  BIT     #20000,@SWR     ;BRANCH IF ERROR
3494  016352  001055                  BNE     LTG1X           ;PRINTOUT DESIRED
3495  016354  005737  000614  LTG1A:  TST     HDRFL           ;SEE IF PRINT HEADER
3496  016360  001004                  BNE     LTG1B           ;IF NOT: BR
3497  016362  013704  000616          MOV     EMADDR,R4
3498  016366  004737  017760          JSR     PC,TTOUT        ;PRINT HEADER
3499  016372  012737  000001  000614  LTG1B:  MOV     #1,HDRFL        ;SET FLAG
3500  016400  013704  000666          MOV     ERADD,R4
3501  016404  004737  017760          JSR     PC,TTOUT        ;PRINT ERROR CODE
3502  016410  005737  000732          TST     PEXFL           ;SEE IF PRINT EXPT RCVD
3503  016414  001016                  BNE     LTG1T           ;IF NOT: BR
3504  016416  012704  021755          MOV     #MSG12,R4
3505  016422  004737  017760          JSR     PC,TTOUT        ;PRINT EXPT TAG
3506  016426  010103                  MOV     R1,R3
3507  016430  004737  020172          JSR     PC,OCTP         ;PRINT EXPT
3508  016434  012704  021764          MOV     #MSG13,R4
3509  016440  004737  017760          JSR     PC,TTOUT        ;PRINT RCVD TAG
3510  016444  010203                  MOV     R2,R3
3511  016446  004737  020172          JSR     PC,OCTP         ;PRINT RCVD
3512  016452  032777  004000  162110  LTG1T:  BIT     #4000,@SWR
3513  016460  001010                  BNE     LTG1C
3514  016462  012704  022026          MOV     #MSG16,R4
3515  016466  004737  017760          JSR     PC,TTOUT
3516  016472  013703  000676          MOV     ITCNT,R3
3517  016476  004737  020172          JSR     PC,OCTP         ;PRINT ITERATION
3518  016502  005237  000662  LTG1C:  INC     PFLG
3519  016506  000240          LTG1X:  NOP
3520  016510  005777  162054          TST     @SWR
3521  016514  100001                  BPL     LTG1X1          ;IF NOT STOP ON ERROR: BR
3522  016516  000000                  HALT
3523  016520  005737  000662  LTG1X1: TST     PFLG
3524  016524  001004                  BNE     LTG1XX          ;IF HAVE PRINTED: BR
3525  016526  032777  020000  162034  BIT     #20000,@SWR
3526  016534  001707                  BEQ     LTG1A
3527  016536  000240          LTG1XX: NOP
3528  016540  005037  000732          CLR     PEXFL           ;CLEAR EXPT RCVD FLAG
3529  016544  000137  016550          JMP     SCOPE           ;GO TO SCOPE
3530
3531
3532                                        ;SCOPE LOOP ON ERROR SUBROUTINE**********
3533
3534  016550  000240          SCOPE:  NOP
3535  016552  032777  040000  162010  BIT     #40000,@SWR     ;SEE IF LOOP ON ERROR
3536  016560  001001                  BNE     1$              ;IF SO: BR
3537  016562  000207                  RTS     PC              ;ELSE EXIT
3538  016564  000240          1$:     NOP
3539  016566  005726                  TST     (SP)+           ;RESET STACK
3540  016570  000240                  NOP
3541  016572  000240                  NOP
3542  016574  000177  160106          JMP     @SCOPE          ;LOOP ON ERROR
3543
```

```
                                                   ;TEST ITERATION SUBROUTINE**********

016600  032777  004000  161762  ITER:   BIT     #4000,@SWR          ;SEE IF ITERATIONS
016606  001403                          BEQ     2$                  ;IF SO: BR
016610  005037  000676          1$:     CLR     ITCNT               ;CLEAR ITERATION COUNTER
016614  000207                          RTS     PC                  ;ELSE EXIT
016616  005737  001014          2$:     TST     PCNTR               ;NO SUBTEST ITERATIONS ON FIRST PASS
016622  001772                          BEQ     1$
016624  005237  000676                  INC     ITCNT               ;BUMP COUNTER
016630  023737  000676  000602          CMP     ITCNT,ITAMT         ;SEE IF DONE ALL
016636  001764                          BEQ     1$                  ;IF SO: BR
016640  005726                          TST     (SP)+               ;RESET STACK
016642  017700  162042                  MOV     @ITPTR,R0           ;SET ITERATION POINTER
016646  000110                          JMP     (R0)                ;GO ITERATE

                                                   ;MANUAL INTERVENTION INHIBIT*********

016650  000240                  INMT:   NOP
016652  012704  022617                  MOV     #MSG43,R4
016656  004737  017760                  JSR     PC,TTOUT            ;GO PRINT INHIB MSG
016662  000000                          HALT
016664  000137  002530                  JMP     TSC02               ;RETURN TO SCHED


                                                   ;NON-STANDARD MODE TEST HANDLER

016670  010046                  NOST:   MOV     R0,-(SP)            ;*SAVE R0
016672  012700  000240                  MOV     #240,R0             ;*SET UP INDEX
016676  013760  001326  001056          MOV     TADX,TSTTBL(R0)     ;*
016704  005720                          TST     (R0)+
016706  012737  000047  001330          MOV     #47,TLAST           ;*SET END OF TEST
016714  013760  001330  001056          MOV     TLAST,TSTTBL(R0)        ;*SET LAST TEST NUMBER
016722  012600                          MOV     (SP)+,R0            ;*RESTORE R0
016724  000207                          RTS     PC                  ;*RETURN
```

```

                                                ;INITIALIZE SUBROUTINE**********

016726  000240                          INIT1:  NOP
016730  012777  000040  161562                  MOV     #40,@CS                 ;INIT
016736  013777  000620  161554          INIT2:  MOV     DRVN,@CS                ;SELECT DRIVE
016744  013777  000660  161570                  MOV     SLVN,@TC                ;SELECT SLAVE
016752  000207                                  RTS     PC                      ;RETURN

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

                                        ;SET SLAVE IN NRZ OFF BOT
016754  013746  000772                  INIT3:  MOV     UDES,-(SP)              ;SAVE TEST'S UNIT DESCRIPTION
016760  012737  001400  000772                  MOV     #1400,UDES              ;SET UNIT DESCRIPTION = NRZ
016766  000410                                  BR      INIT5                   ;GO TO INIT5 ROUTINE

                                        ;SET SLAVE IN PE OFF BOT
016770  013746  000772                  INIT4:  MOV     UDES,-(SP)              ;SAVE TEST'S UNIT DESRIPTION
016774  012737  002000  000772                  MOV     #2000,UDES              ;SET UNIT DESRIPTION = PE
017002  000402                                  BR      INIT5                   ;GO DO IT

                                        ;THIS ROUTINE IS ENTERED AT INIT WHEN THE CALLER HAS SETUP UDES.
                                        ;IT IS ENTERED AT INIT5 WHEN EITHER INIT3 OR INIT4 HAS SET UP UDES.
017004  013746  000772                  INIT:   MOV     UDES,-(SP)              ;SAVE TEST'S UNIT DESRIPTION
017010  012777  000040  161502          INIT5:  MOV     #40,@CS                 ;INIT CONTROLLER
017016  013777  000620  161474                  MOV     DRVN,@CS                ;SELECT TM03 DRIVE
017024  013777  000660  161510                  MOV     SLVN,@TC                ;SELECT TE16 SLAVE
017032  013746  000772                          MOV     UDES,-(SP)              ;GET SLAVE DESCRIPTION
017036  042716  174377                          BIC     #174377,(SP)            ;CLEAR ALL BUT DENSITY SELECT BITS
017042  022726  001400                          CMP     #1400,(SP)+             ;BRANCH IF REQUESTING PE MODE
017046  001005                                  BNE     1$
017050  032777  000040  161444                  BIT     #40,@DS                 ;BRANCH IF SLAVE IS IN NRZ MODE
017056  001420                                  BEQ     4$                      ;(PES = 0)
017060  000404                                  BR      2$
017062  032777  000040  161432          1$:     BIT     #40,@DS                 ;BRANCH IF SLAVE IS IN PE MODE
017070  001013                                  BNE     4$
017072  012777  000007  161410          2$:     MOV     #7,@C1                  ;REWIND SLAVE
017100  032777  000200  161414          20$:    BIT     #200,@DS                ;WAIT FOR READY
017106  001774                                  BEQ     20$
017110  032777  020000  161404          3$:     BIT     #20000,@DS              ;WAIT UNTIL PIP CLEARS
017116  001374                                  BNE     3$
017120  053777  000772  161414          4$:     BIS     UDES,@TC                ;LOAD SLAVE DESRIPTION
017126  032777  000002  161366                  BIT     #2,@DS                  ;BRANCH IF NOT AT BOT
017134  001407                                  BEQ     6$
017136  012777  000025  161344                  MOV     #25,@C1                 ;ERASE TO GET OFF BOT
017144  032777  000200  161350          5$:     BIT     #200,@DS                ;WAIT FOR READY
017152  001774                                  BEQ     5$
017154  032777  000020  161340          6$:     BIT     #20,@DS ;WAIT FOR SETTLEDOWN TO CLEAR
017162  001374                                  BNE     6$
017164  012777  000011  161316                  MOV     #11,@C1                 ;RESET DRIVE
017172  012637  000772                          MOV     (SP)+,UDES              ;RESTORE UNIT DESRIPTION
017176  000207                                  RTS     PC                      ;RETURN

```

```
3636
3637
3638                                        ;MANUAL INSTRUCTION SUBROUTINE**********
3639
3640  017200  000240              INST:     NOP
3641  017202  004737  017760              JSR     PC,TTOUT        ;PRINT INSTRUCTION
3642  017206  012704  027122              MOV     #MMSG0,R4
3643  017212  004737  017760              JSR     PC,TTOUT        ;PRINT REPLY
3644  017216  012705  000674              MOV     #TEMP3,R5
3645  017222  012701  000001              MOV     #1,R1
3646  017226  012702  177777              MOV     #-1,R2
3647  017232  012703  000000              MOV     #0,R3
3648  017236  004737  017436              JSR     PC,TTR          ;AWAIT REPLY
3649  017242  000240                      NOP
3650  017244  000207                      RTS     PC              ;EXIT
3651
3652                                        ;MAG TAPE INTERRUPT HANDLER**********
3653
3654  017246  000240              MTINT:    NOP
3655  017250  013716  000664              MOV     RTRN,(SP)       ;SET RETURN FROM INTERUPT ADDRESS
3656  017254  000002                      RTI                     ;RETURN
3657
3658                                        ;TTY INTERRUPT HANDLER**********
3659
3660  017256  017746  161312    TTINT:    MOV     @TKB,-(SP)      ;GET CHARACTER
3661  017262  042716  000200              BIC     #200,(SP)       ;CLEAR PARITY BIT
3662  017266  122716  000003              CMPB    #3,(SP)         ;BRANCH IF NOT CONTROL C
3663  017272  001010                      BNE     1$
3664  017274  005737  00142c              TST     CHNFLG          ;INHIBIT ^C IF IN CHAIN MODE
3665  017300  001005                      BNE     1$
3666  017302  005077  161260              CLR     @PSW            ;CLEAR PSW
3667  017306  000005                      RESET
3668  017310  000137  000200              JMP     @#200           ;RESTART
3669  017314  122716  000001      1$:     CMPB    #1,(SP)         ;BRANCH IF NOT ^A
3670  017320  001017                      BNE     2$
3671  017322  022737  000176  000570      CMP     #SWREG,SWR      ;BRANCH IF USING HARWARE SWR
3672  017330  001016                      BNE     3$
3673  017332  012737  177570  000570      MOV     #177570,SWR     ;INVOKE HARDWARE SWR
3674  017340  004737  020716              JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
3675  017344  012704  023614              MOV     #MSG63,R4       ;TYPE 'HARDWARE SWR IN USE'
3676  017350  004737  017760              JSR     PC,TTOUT
3677  017354  004737  020740              JSR     PC,.RESTORE
3678  017360  122716  000007      2$:     CMPB    #7,(SP)         ;BRANCH IF NOT ^G
3679  017364  001006                      BNE     4$
```

```
 3681  017366  012737  000176  000570  3$:     MOV     #SWREG,SWR      ;INVOKE SOFTWARE SWR
 3682  017374  004737  020620                  JSR     PC,GTSWR        ;GET SWITCHES
 3683  017400  000414                          BR      6$
 3684  017402  122716  000023          4$:     CMPB    #23,(SP)        ;SEE IF ^S
 3685  017406  001004                          BNE     5$              ;BRANCH IF NOT
 3686  017410  112737  000377  001332          MOVB    #377,$CNTRLS    ;SET XOFF FLAG
 3687  017416  000405                          BR      6$
 3688  017420  122716  000021          5$:     CMPB    #21,(SP)        ;SEE IF ^Q
 3689  017424  001002                          BNE     6$              ;BRANCH IF NOT
 3690  017426  105037  001332                  CLRB    $CNTRLS
 3691  017432  005726                  6$:     TST     (SP)+           ;POP CHARACTER OFF STACK
 3692  017434  000002                          RTI                     ;RETURN
 3693
```

```
                                        ;*********************************************************
                                        ;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
                                        ;*********************************************************

017436  010146              TTR:    MOV     R1,-(SP)        ;SAVE CHARACTER COUNT
017440  011601              10$:    MOV     (SP),R1         ;RESTORE CHARACTER COUNT (FOR TO)
017442  005037  000670            CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
017446  005000                    CLR     R0
017450  004737  017716      1$:   JSR     PC,TTIN         ;GO READ CHARACTER
017454  122737  000003  000612    CMPB    #3,TIB          ;BRANCH IF NOT ^C
017462  001003                    BNE     11$
017464  000005                    RESET                   ;RESET
017466  000137  000200            JMP     @#200           ;RESTART PROGRAM
017472  122737  000015  000612 11$:  CMPB    #15,TIB         ;SEE IF CR
017500  001004                    BNE     2$              ;IF NOT: BR
017502  005737  000670            TST     TEMP1           ;SEE IF FIRST CHARACTER
017506  001471                    BEQ     9$              ;IF SO: BR
017510  000457                    BR      6$
017512  122737  000025  000612 2$:   CMPB    #25,TIB         ;BRANCH IF NOT CONTROL U
017520  001005                    BNE     21$
017522  012704  023542            MOV     #MSG59,R4       ;TYPE <CR><LF>
017526  004737  017760            JSR     PC,TTOUT
017532  000742                    BR      10$
017534  122737  000177  000612 21$:  CMPB    #177,TIB        ;BRANCH IF NOT 'RUBOUT'
017542  001012                    BNE     3$
017544  000241                    CLC
017546  006000                    ROR     R0              ;REMOVE LAST CHAR
017550  006200                    ASR     R0
017552  006200                    ASR     R0
017554  012704  023544            MOV     #MSG60,R4       ;TYPE '\
017560  004737  017760            JSR     PC,TTOUT
017564  005201                    INC     R1              ;DECREMENT CHARS RECEIVED COUNT
017566  000730                    BR      1$
017570  122737  000060  000612 3$:   CMPB    #'0,TIB         ;SEE IF CHAR IS LESS THAN 0
017576  101402                    BLOS    4$              ;IF NOT: BR
017600  000137  017676            JMP     TINER           ;ELSE GO TO ERROR
017604  122737  000070  000612 4$:   CMPB    #'7+1,TIB       ;SEE IF CHAR IS GREATER THAN 7
017612  101002                    BHI     5$              ;IF NOT: BR
017614  000137  017676            JMP     TINER           ;ELSE GO TO ERROR
017620  005237  000670        5$:   INC     TEMP1           ;SET FIRST CHARACTER FLAG
017624  006300                    ASL     R0
017626  006300                    ASL     R0              ;SHIFT 3 LEFT
017630  006300                    ASL     R0
```

```
CZTEAE0 IN03 TE16-1077 CTL I    MACY11 30(1046)  06-APR-84  09:48  PAGE 82-1
CZTEAE.P11     06-APR-84 09:45                                          SEQ 0098

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

```
                                        ;TTY ENTRY ERROR SUBROUTINE**********
017676  012704  022557        TINER:  MOV     #MSG40,R4
017702  004737  017760                JSR     PC,TTOUT        ;PRINT?
017706  005726                        TST     (SP)+           ;POP CHAR COUNT OFF STACK
017710  162716  000020                SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
017714  000207                        RTS     PC              ;REDO VALUE ENTRY

                                        ;TTY READ SUBROUTINE**********
017716  005277  160650        TTIN:   INC     @TKS
017722  105777  160644        1$:     TSTB    @TKS
017726  100375                        BPL     1$
017730  017737  160640  000612        MOV     @TKB,TIB
017736  042737  000200  000612        BIC     #200,TIB                ;STRIP PARITY BIT
017744  013737  000612  000610        MOV     TIB,TOB                 ;MOVE CHAR TO TTY OUTPUT BFR
017752  004737  020060                JSR     PC,TOG                  ;AND ECHO IT
017756  000207                        RTS     PC

                                        ;TTY OUTPUT SUBROUTINE**********
017760  112437  000610        TTOUT:  MOVB    (R4)+,TOB
017764  122737  000043  000610        CMPB    #43,TOB
017772  001472                        BEQ     TEX
017774  122737  000045  000610        CMPB    #45,TOB
020002  001403                        BEQ     1$
020004  004737  020060                JSR     PC,TOG
020010  000763                        BR      TTOUT
020012  112737  000015  000610  1$:   MOVB    #15,TOB
020020  004737  020060                JSR     PC,TOG
020024  012703  000004                MOV     #4,R3
020030  005037  000610        2$:     CLR     TOB
020034  004737  020060                JSR     PC,TOG
020040  005303                        DEC     R3
020042  001372                        BNE     2$                      ;DO 4 TIMES
020044  112737  000012  000610        MOVB    #12,TOB
020052  004737  020060                JSR     PC,TOG
020056  000740                        BR      TTOUT
```

```
3806
3807  020060  105777  160506          TOG:    TSTB    @TKS            ;SEE IF INPUT AT KEYBOARD
3808  020064  100024                  BPL     3$              ;IF SO, THEN
3809  020066  117737  160502  001333          MOVB    @TKB,$CNTRLS+1  ;MOVE CHARACTER AND
3810  020074  142737  000200  001333          BICB    #200,$CNTRLS+1  ;MASK OFF PARITY BIT.
3811  020102  122737  000023  001333          CMPB    #23,$CNTRLS+1   ;SEE IF CHARACTER IS XOFF
3812  020110  001004                  BNE     2$              ;IF XOFF, THEN
3813  020112  112737  000377  001332          MOVB    #377,$CNTRLS    ;SET XOFF FLAG
3814  020120  000757                  BR      TOG
3815  020122  122737  000021  001333  2$:     CMPB    #21,$CNTRLS+1   ;SEE IF CHARACTER IS XON
3816  020130  001002                  BNE     3$              ;IF SO THEN
3817  020132  105037  001332                  CLRB    $CNTRLS         ;CLEAR XOFF FLAG
3818  020136  105737  001332          3$:     TSTB    $CNTRLS         ;SEE IF IN XOFF MODE
3819  020142  100746                  BMI     TOG             ;IF NOT THEN
3820  020144  105777  160426                  TSTB    @TPS            ;CHECK IF PRINTER READY
3821  020150  100343                  BPL     TOG             ;
3822  020152  113777  000610  160420          MOVB    TOB,@TPB
3823  020160  000207          TEX:    RTS     PC              ;RETURN
3824
3825
3826                                  ;OCTAL OUTPUT SUBROUTINE**********
3827
3828  020162  012737  000001  020412  OCTPE:  MOV     #1,OFL
3829  020170  000402                  BR      OCTPE1
3830  020172  005037  020412  OCTP:   CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
3831  020176  010304          OCTPE1: MOV     R3,R4           ;SEE IF NUMBER IS ZERO
3832  020200  001006                  BNE     OCTP0           ;IF NOT ZERO: BR
3833  020202  005737  020412                  TST     OFL             ;SEE IF PRINT ALL 0
3834  020206  001003                  BNE     OCTP0           ;IF SO: BR
3835  020210  004737  020372                  JSR     PC,OCTPG1       ;ELSE PRINT ZERO
3836  020214  000447                  BR      OCTP3           ;SPACE AND EXIT
3837  020216  032704  100000  OCTP0:  BIT     #100000,R4      ;SEE IF MSD = 1
3838  020222  001405                  BEQ     OCTP1           ;IF NOT: BR
3839  020224  012704  000001                  MOV     #1,R4
3840  020230  004737  020350                  JSR     PC,OCTPG        ;PRINT 1
3841  020234  000403                  BR      OCTP2
3842  020236  005004          OCTP1:  CLR     R4
3843  020240  004737  020350                  JSR     PC,OCTPG        ;PRINT 0
3844  020244  010304          OCTP2:  MOV     R3,R4
3845  020246  006004                  ROR     R4
3846  020250  006004                  ROR     R4
3847  020252  006004                  ROR     R4              ;POSITION DIGIT
3848  020254  006004                  ROR     R4
3849  020256  000304                  SWAB    R4
3850  020260  004737  020350                  JSR     PC,OCTPG        ;PRINT DIGIT 2
3851  020264  010304                  MOV     R3,R4
3852  020266  006004                  ROR     R4
3853  020270  000304                  SWAB    R4
3854  020272  004737  020350                  JSR     PC,OCTPG        ;PRINT DIGIT 3
3855  020276  010304                  MOV     R3,R4
3856  020300  006104                  ROL     R4
3857  020302  006104                  ROL     R4
3858  020304  000304                  SWAB    R4
3859  020306  004737  020350                  JSR     PC,OCTPG        ;PRINT DIGIT 4
3860  020312  010304                  MOV     R3,R4
3861  020314  006004                  ROR     R4
```

```
 3862  020316  006004                        ROR     R4
 3863  020320  006004                        ROR     R4
 3864  020322  004737  020350                JSR     PC,OCTPG
 3865  020326  010304                        MOV     R3,R4
 3866  020330  004737  020350                JSR     PC,OCTPG        ;PRINT DIGIT 5
 3867  020334  012737  000240  000610  OCTP3: MOV     #240,TOB
 3868  020342  004737  020060                JSR     PC,TOG          ;PRINT SPACE
 3869  020346  000207                        RTS     PC              ;EXIT
 3870
 3871  020350  042704  177770          OCTPG: BIC     #177770,R4
 3872  020354  001004                        BNE     OCTPG0
 3873  020356  005737  020412                TST     OFL
 3874  020362  001001                        BNE     OCTPG0
 3875  020364  000207                        RTS     PC
 3876
 3877  020366  005237  020412          OCTPG0: INC     OFL
 3878  020372  052704  000260          OCTPG1: BIS     #260,R4
 3879  020376  010437  000610                MOV     R4,TOB
 3880  020402  004737  020060                JSR     PC,TOG
 3881  020406  010304                        MOV     R3,R4
 3882  020410  000207                        RTS     PC
 3883  020412  000000                OFL:    0                       ;FIRST CHAR FLAG
 3884
```

```
                                                ;DATA CHARACTER OUTPUT SUBROUTINE************

020414  012704  000010          DOUT:   MOV     #10,R4                  ;SET NUMBER TO PRINT
020420  110337  000610                  MOVB    R3,TOB
020424  105777  160146          1$:     TSTB    @TPS
020430  100375                          BPL     1$
020432  132737  000200  000610          BITB    #200,TOB
020440  001404                          BEQ     2$
020442  012777  000061  160130          MOV     #061,@TPB
020450  000403                          BR      3$
020452  012777  000060  160120  2$:     MOV     #060,@TPB
020460  006337  000610          3$:     ASL     TOB
020464  005304                          DEC     R4
020466  001356                          BNE     1$
020470  000207                          RTS     PC

020472  013703  000674          DOUTD:  MOV     TEMP3,R3
020476  000303                          SWAB    R3
020500  004737  020414                  JSR     PC,DOUT
020504  013703  000674                  MOV     TEMP3,R3
020510  004737  020414                  JSR     PC,DOUT
020514  000207                          RTS     PC

                                                ;TE16 SERIAL NUMBER PRINT SUBROUTINE**********

020516  010304                  SNPT:   MOV     R3,R4
020520  000304                          SWAB    R4
020522  006004                          ROR     R4
020524  006004                          ROR     R4
020526  006004                          ROR     R4
020530  006004                          ROR     R4              ;GET FIRST DIGIT
020532  004737  020574                  JSR     PC,SNPG         ;PRINT
020536  010304                          MOV     R3,R4
020540  000304                          SWAB    R4              ;GET SECOND DIGIT
020542  004737  020574                  JSR     PC,SNPG         ;PRINT
020546  010304                          MOV     R3,R4
020550  006004                          ROR     R4
020552  006004                          ROR     R4
020554  006004                          ROR     R4
020556  006004                          ROR     R4
020560  004737  020574                  JSR     PC,SNPG         ;PRINT THIRD DIGIT
020564  010304                          MOV     R3,R4
020566  004737  020574                  JSR     PC,SNPG         ;PRINT FOURTH DIGIT
020572  000207                          RTS     PC              ;EXIT
020574  012737  000260  000610  SNPG:   MOV     #260,TOB        ;SET BASE = 0
020602  042704  177760                  BIC     #177760,R4      ;MASK DIGIT
020606  050437  000610                  BIS     R4,TOB          ;SET ASCII
020612  004737  020060                  JSR     PC,TOG          ;TYPE DIGIT
020616  000207                          RTS     PC              ;RETURN
```

```
3937
3938                                    ;ROUTINE TO LOAD CONTENTS OF SOFTWARE SWITCH REGISTER.
3939                                    ;IF A CONTROL G (^G) IS TYPED THE SOFTWARE SWITCH REGISTER IS LOADED
3940  020620  022737  000176  00057C  GTSWR:  CMP     #SWREG,SWR      ;BRANCH IF SOFTWARE SWR
3941  020626  001032                        BNE     1$              ;NOT INVOKED
3942  020630  004737  020716                JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
3943  020634  012704  027473                MOV     #$MSWR,R4       ;TYPE 'SWR = '
3944  020640  004737  017760                JSR     PC,TTOUT
3945  020644  017703  157720                MOV     @SWR,R3         ;GET CURRENT VALUE
3946  020650  004737  020162                JSR     PC,OCTPE        ;AND TYPE IT
3947  020654  012704  027503                MOV     #$MNEW,R4       ;ASK FOR NEW VALUE
3948  020660  004737  017760                JSR     PC,TTOUT
3949  020664  013705  000570                MOV     SWR,R5          ;NEW VALUE WILL BE RETURNED IN (R5)
3950  020670  012701  000007                MOV     #7,R1           ;LIMIT TO 7 CHARACTERS
3951  020674  012702  177777                MOV     #177777,R2      ;LIMIT RESPONSE TO BETWEEN
3952  020700  012703  000000                MOV     #0,R3           ;0 AND 177777
3953  020704  004737  017436                JSR     PC,TTR          ;GET RESPONSE
3954  020710  004737  020740                JSR     PC,.RESTORE     ;RESTORE REGISTERS
3955  020714  000207                1$:     RTS     PC              ;RETURN TO CALLER
3956                                    ;;ROUTINE TO SAVE REGISTERS ON THE STACK
 (1)  020716  010546                .SAVE:  MOV     %5,-(SP)        ;;R5 IS SAVED AT 12(SP)
 (1)  020720  010446                        MOV     %4,-(SP)        ;;R4 IS SAVED AT 10(SP)
 (1)  020722  010346                        MOV     %3,-(SP)        ;;R3 IS SAVED AT 6(SP)
 (1)  020724  010246                        MOV     %2,-(SP)        ;;R2 IS SAVED AT 4(SP)
 (1)  020726  010146                        MOV     %1,-(SP)        ;;R1 IS SAVED AT 2(SP)
 (1)  020730  010046                        MOV     %0,-(SP)        ;;R0 IS SAVED AT (SP)
 (1)  020732  016646  000014                MOV     14(SP),-(SP)    ;;PUSH RETURN PC ON THE STACK
 (1)  020736  000207                        RTS     PC              ;;RETURN TO CALLER
 (1)
3957                                    ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
 (1)  020740  012666  000014        .RESTORE:MOV    (SP)+,14(SP)    ;;STORE RETURN PC ON STACK
 (1)  020744  012600                        MOV     (SP)+,%0
 (1)  020746  012601                        MOV     (SP)+,%1
 (1)  020750  012602                        MOV     (SP)+,%2
 (1)  020752  012603                        MOV     (SP)+,%3
 (1)  020754  012604                        MOV     (SP)+,%4
 (1)  020756  012605                        MOV     (SP)+,%5
 (1)  020760  000207                        RTS     PC              ;;RETURN
 (1)
3958
3959                                    ;MESSAGE TABLE**********
3960
3961  020762  022445  046524  031460  MSG1:   .ASCII  '%%TM03-TE16/TU77 CONTROL LOGIC TE..  PART 1 (CZTEAE0)';++8
      020770  052055  030505  027466
      020776  052524  033467  041440
      021004  047117  051124  046117
      021012  046040  043517  041511
      021020  052040  051505  026524
      021026  050040  051101  020124
      021034  020111  041450  052132
      021042  040505  030105     051
3962  021047     045  025052  040452          .ASCII  /%***ASSURE TAPE IS AT BOT***/
      021054  051523  051125  020105
      021062  040524  042520  044440
      021070  020123  052101  041040
      021076  052117  025052     052
```

```
3963  021103  045     054524  042520          .ASCII  /%TYPE <CR> TO TERMINATE RESPONSE & ^C TO RESTART%#/
      021110  036040  051103  020076
      021116  047524  052040  051105
      021124  044515  040516  042524
      021132  051040  051505  047520
      021140  051516  020105  020046
      021146  041536  052040  020117
      021154  042522  052123  051101
      021162  022524  043
3964  021165  045     051104  053111  MSG2:   .ASCII  /%DRIVE NUMBER OR (CR) WHEN DONE #/
      021172  020105  052516  041115
      021200  051105  047440  020122
      021206  041450  024522  053440
      021214  042510  020116  047504
      021222  042516  021440
3965  021226  022445  047506  020122  MSG2A:  .ASCII  /%%FOR DRIVE ADDRESS TEST;/
      021234  051104  053111  020105
      021242  042101  051104  051505
      021250  020123  042524  052123
      021256  073
3966  021257  045     042440  052116          .ASCII  /% ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
      021264  051105  042440  050130
      021272  020124  051104  053111
      021300  020105  052516  041115
      021306  051105  020054  046101
      021314  020114  052117  042510
      021322  051522  051440  047510
      021330  046125  020104  042502
      021336  047040  047117  042455
      021344  044530  052123  047101
      021352  027124  043
3967  021355  045     047516  026516  MSG3:   .ASCII  /%NON-EXIST DRIVE #/
      021362  054105  051511  020124
      021370  051104  053111  020105
      021376  043
3968  021377  045     044122  042040  MSG4:   .ASCII  /%RH DETECTED #/
      021404  052105  041505  042524
      021412  020104  043
3969  021415  045     046524  031460  MSG5:   .ASCII  /%TM03 DETECTED #/
      021422  042040  052105  041505
      021430  042524  020104  043
3970  021435  105     050130  026524  MSG6:   .ASCII  /EXPT-NOT RECVD#/
      021442  047516  020124  042522
      021450  053103  021504
3971  021454  041522  042126  047055  MSG7:   .ASCII  /RCVD-NOT EXPT#/
      021462  052117  042440  050130
      021470  021524
3972  021472  051445  040514  042526  MSG8:   .ASCII  /%SLAVE NUMBER OR (CR) WHEN DONE #/
      021500  047040  046525  042502
      021506  020122  051127  024040
      021514  051103  020051  044127
      021522  047105  042040  047117
      021530  020105  043
3973  021533  045     043045  051117  MSG8A:  .ASCII  /%%FOR SLAVE ADDRESS TEST;/
      021540  051440  040514  042526
      021546  040440  042104  042522
```

```
                021554  051523  052040  051505
                021562  035524
3974            021564  020045  047105  042524          ,ASCII  /* ENTER EXPT SLAVE NUMBER, ALL OTHERS SHOULD BE NON EXISTANT,0/
                021572  020122  054105  052120
                021600  051440  040514  042526
                021606  047040  046525  042502
                021614  026122  040440  046114
                021622  047440  044124  051105
                021630  020123  044123  052517
                021636  042114  041040  020105
                021644  047516  026516  054105
                021652  051511  040524  052116
                021660  021456
3975            021662  047045  047117  042455  MSG9:   ,ASCII  /*NON EXIST SLAVE 0/
                021670  044530  052123  051440
                021676  040514  042526  021440
3976            021704  051045  040505  020104  MSG10:  ,ASCII  /*READ CONT BUS PAR 0/
                021712  047503  052116  041040
                021720  051525  050040  051101
                021726  021440
3977            021730  053445  044522  042524  MSG11:  ,ASCII  /*WRITE CONT BUS PAR 0/
                021736  041440  047117  020124
                021744  052502  020123  040520
                021752  020122     043
3978            021755     040  054105  052120  MSG12:  ,ASCII  / EXPT 0/
                021762  021440
3979            021764  051040  053103  020104  MSG13:  ,ASCII  / RCVD 0/
                021772     043
3980            021773     045  051115  041040  MSG14:  ,ASCII  /*MR BITS 4 00/
                022000  052111  020123  026464
                022006  021460
3981            022010  046445  020122  044502  MSG15:  ,ASCII  /*MR BITS 15 70/
                022016  051524  030440  026465
                022024  021467
3982            022026  044445  042524  035122  MSG16:  ,ASCII  /*ITER: 0/
                022034  021440
3983            022036  052045  020103  044502  MSG18:  ,ASCII  /*TC BITS 15 0 0
                022044  051524  030440  026462
                022052  020060     043
3984            022055     045  041506  041040  MSG19:  ,ASCII  /*FC BITS 15 0 0
                022062  052111  020123  032461
                022070  030055  021440
3985            022074  043045  047125  041440  MSG20:  ,ASCII  /*RUN CODE BITS 5 1 OF C1 0
                022102  042117  020105  044502
                022110  051524  052440  030455
                022116  047440  020106  030503
                022124  021440
3986            022126  043445  020117  044502  MSG21:  ,ASCII  /*GO BIT NOT CORRECT AT START 0
                022134  020124  047516  020124
                022142  047503  051122  041505
                022150  020124  052101  051440
                022156  040524  052122  021440
3987            022164  043445  020117  044502  MSG22:  ,ASCII  /*GO BIT NOT SET 0
                022172  020124  047516  020124
                022200  042522  020124     043
3988            022205     045  047507  041040  MSG23:  ,ASCII  /*GO BIT NOT RESET BY INIT 0
```

```
                022210  052111  047040  052117
                022220  051040  051505  052105
                022226  041040  020131  047111
                022234  052111  021440
 3989           022240  042045  054522  047040  MSG24:  .ASCII  /*DRY NOT SET BY INIT 0/
                022246  052117  051440  052105
                022254  041040  020131  047111
                022262  052111  021440
 3990           022266  042045  054522  047040  MSG25:  .ASCII  /*DRY NOT RESET BY GO*10/
                022274  052117  051040  051505
                022302  052105  041040  020131
                022310  047507  030475     043
 3991           022315     045  051104  020131  MSG25A: .ASCII  /*DRY NOT SET BY GO*00/
                022322  047516  020124  042523
                022330  020124  054502  043440
                022336  036517  021460
 3992           02234.  047045  020117  047111  MSG26:  .ASCII  /*NO INTERRUPT RETURNED0/
                022350  042524  051122  050125
                022356  020124  042522  052524
                022364  047122  042105     043
 3993           022371     045  040502  020104  MSG27:  .ASCII  /*BAD STATUS0/
                022376  052123  052101  051525
                022404     043
 3994           022405     040  047123  020072  MSG30:  .ASCII  / SN: 0/
                022412     043
 3995           022413     045  051105  020122  MSG31:  .ASCII  /*ERR NOT SET 0/
                022420  047516  020124  042523
                022426  020124     043
 3996           022431     045  052101  020101  MSG32:  .ASCII  /*ATA NOT SET 0/
                022436  047516  020124  042523
                022444  020124     043
 3997           022447     045  051501  041040  MSG33:  .ASCII  /*AS BIT NOT SET 0/
                022454  052111  047040  052117
                022462  051440  052105  021440
 3998           022470  051445  020103  047516  MSG34:  .ASCII  /*SC NOT SET 0/
                022476  020124  042523  020124
                022504     043
 3999           022505     045  051124  020105  MSG35:  .ASCII  /*TRE NOT SET 0/
                022512  047516  020124  042523
                022520  020124     043
 4000           022523     045  046123  020101  MSG36:  .ASCII  /*SCA NOT SET 0/
                022530  047516  020124  042523
                022536  020124     043
 4001           022541     045  051523  020103  MSG37:  .ASCII  /*SSC NOT SET 0/
                022546  047516  020124  042523
                022554  020124     043
 4002           022557     040  020077     043  MSG40:  .ASCII    / 0/
 4003           022563     045  042445  042116  MSG41:  .ASCII  /**END OF PASS 0/
                022570  047440  020106  040520
                022576  051523  021440
 4004           022602  042045  040505  020104  MSG42:  .ASCII  /*DEAD TRACK 0/
                022610  051124  041501  020113
                022616     043
 4005           022617     045  046445  047101  MSG43:  .ASCII  /**MANUAL TESTS (14 17) INHIBITED; HALT*
                022624  040525  020114  042524
                022632  052123  020123  050450
```

```
CZTEAE.MO3-TE16-TU77-CTL-I    MACY11 30(1046)  06-APR-84  09:48  PAGE 86-4
CZTEAE.P11    06-APR-84 09:45                                          SEQ 0107

        022640  026464  033461  020051
        022646  047111  044510  044502
        022654  042524  035104  044040
        022662  046101  022524
 4006   022666  042522  042523  042514          .ASCII  /RESELECT AND PRESS CONTINUE*0/
        022674  052103  040440  042116
        022702  050040  042522  051523
        022710  041440  047117  044524
        022716  052516  022505     043
 4007   022723     045  042522  044507  MSG44:  .ASCII  /*REGISTER START: 0/
        022730  052123  051105  051440
        022736  040524  052122  020072
        022744     043
 4008   022745     045  042526  052103  MSG45:  .ASCII  /*VECTOR ADDRESS: 0/
        022752  051117  040440  042104
        022760  042522  051523  020072
        022766     043
 4009   022767     045  051503  020061  MSG46:  .ASCII  /*CS1     WC      BA      FC      CS2     DS      ER      AS/
        022774  020040  053440  020103
        023002  020040  020040  040502
        023010  020040  020040  043040
        023016  020103  020040  020040
        023024  051503  020062  020040
        023032  042040  020123  020040
        023040  020040  051105  020040
        023046  020040  040440     123
 4010   023053     040  020040  020040          .ASCII  /      MR      TC*0/
        023060  051115  020040  020040
        023066  052040  022503     043
 4011   023073     045  047516  020124  MSG47:  .ASCII  /*NOT RESET BY DRIVE CLEAR0/
        023100  042522  042523  020124
        023106  054502  042040  044522
        023114  042526  041440  042514
        023122  051101     043
 4012   023125     045  046101  044120  MSG50:  .ASCII  /*ALPHA NOT SET0/
        023132  020101  047516  020124
        023140  042523  021524
 4013   023144  052445  042516  050130  MSG51:  .ASCII  /*UNEXPECTED ERROR BITS0/
        023152  041505  042524  020104
        023160  051105  047522  020122
        023166  044502  051524     043
 4014   023173     045  040502  020104  MSG53:  .ASCII  /*BAD LRC 0/
        023200  051114  020103     043
 4015   023205     045  040502  020104  MSG54:  .ASCII  /*BAD CK 0/
        023212  045503  021440
 4016   023216  051445  052105  050125  MSG55:  .ASCII  /*SETUP ERROR: CHECK WRAP 0 WITH CONTROL LOGIC TEST 11 TEST 10/
        023224  042440  051122  051117
        023232  020072  044103  041505
        023240  020113  051127  050101
        023246  030040  053440  052111
        023254  020110  047503  052116
        023262  047522  020114  047514
        023270  044507  020103  042524
        023276  052123  044440  020111
        023304  042524  052123  033440
        023312     043
```

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

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

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

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

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

```
                026316  020103  042524  052123
                026324  032040  035060  050040
                026332  020105  040524  042520
                026340  046440  051101  024113
                026346  034115  031471  024462
                026354     043
4066            026355     045  046045  043517  MSLT41: .ASCII  /**LOGIC TEST 41: NRZ TAPE MARK (M8934)#/
                026362  041511  052040  051505
                026370  020124  030464  020072
                026376  051116  020132  040524
                026404  042520  046440  051101
                026412  020113  046450  034470
                026420  032063  021451
4067            026424  022445  047514  044507  MSLT42: .ASCII  /**LOGIC TEST 42: CRC(M8934)#/
                026432  020103  042524  052123
                026440  032040  035062  041440
                026446  041522  046450  034470
                026454  032063  021451
4068            026460  022445  047514  044507  MSLT43: .ASCII  /**LOGIC TEST 43: LRC(M8934)#/
                026466  020103  042524  052123
                026474  032040  035063  046040
                026502  041522  046450  034470
                026510  032063  021451
4069            026514  022445  047514  044507  MSLT44: .ASCII  /**LOGIC TEST 44: CORRECTABLE DATA (M8932 M8901)#/
                026522  020103  042524  052123
                026530  032040  035064  041440
                026536  051117  042522  052103
                026544  041101  042514  042040
                026552  052101  020101  046450
                026560  034470  031063  046440
                026566  034470  030460  021451
4070            026574  022445  047514  044507  MSLT45: .ASCII  /**LOGIC TEST 45: INCORRECTABLE DATA (M8932 M8934)#/
                026602  020103  042524  052123
                026610  032040  035065  044440
                026616  041516  051117  042522
                026624  052103  041101  042514
                026632  042040  052101  020101
                026640  046450  034470  031063
                026646  046440  034470  032063
                026654  021451
4071            026656  022445  047514  044507  MSLT46: .ASCII  /**LOGIC TEST 46: PEF(M8932)#/
                026664  020103  042524  052123
                026672  032040  035066  050040
                026700  043105  046450  034470
                026706  031063  021451
4072            026712  022445  047514  044507  MSLT47: .ASCII  /**LOGIC TEST 47: FC OVERFLOW (M8905-1B)#/
                026720  020103  042524  052123
                026726  032040  035067  043040
                026734  020103  053117  051105
                026742  046106  053517  024040
                026750  034115  030071  026465
                026756  041131  021451
4073            026762  022445  047514  044507  MSLT50: .ASCII  /**LOGIC TEST 50: NEF WHEN WRITE PE ON NRZ SLAVE#/
                026770  020103  042524  052123
                026776  032440  035060  047040
                027004  043105  053440  042510
```

```
        027012  020116  051127  052111
        027020  020105  042520  047440
        027026  020116  051116  020132
        027034  046123  053101  021505
4074    027042  022445  047514  044507  MSL151: .ASCII  /**LOGIC TEST 51: NEF WHEN WRITE NRZ ON PE SLAVE#/
        027050  020103  042524  052123
        027056  032440  035061  047040
        027064  043105  053440  042510
        027072  020116  051127  052111
        027100  020105  051116  020132
        027106  047117  050040  020105
        027114  046123  053101  021505
```

M9


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

```
 4117  027751      045  051516  020107  TMS27:  .ASCII  /*NSG */
       027756      043
 4118  027757      045  041506  020105  TMS28:  .ASCII  /*FCE */
       027764      043
 4119  027765      045  051503  021440  TMS29:  .ASCII  /*CS */
 4120  027772  044445  046524  021440  TMS30:  .ASCII  /*ITM */
 4121  030000  047045  043105  021440  TMS31:  .ASCII  /*NEF */
 4122  030006  042045  042524  021440  TMS32:  .ASCII  /*DTE */
 4123  030014  047145  044520  021440  TMS33:  .ASCII  /*OPI */
 4124  030022  053445  044522  042524  TMS33A: .ASCII  /*WRITE OPI */
       030030  047440  044520  021440
 4125  030036  051045  040505  020104  TMS33B: .ASCII  /*READ OPI */
       030044  050117  020111     043
 4126  030051     040  041517  052503  TMS33C: .ASCII  / OCCURED TO SOON*0/
       030056  042522  020104  047524
       030064  051440  047517  022516
       030072     043
 4127  030073     040  041517  052503  TMS33D: .ASCII  / OCCURRED TO LATE*0/
       030100  051122  042105  052040
       030106  020117  040514  042524
       030114  021445
 4128  030116  043040  044501  042514  TMS33E: .ASCII  / FAILED TO SET*0/
       030124  020104  047524  051440
       030132  052105  021445
 4129  030136  052445  051516  021440  TMS34:  .ASCII  /*UNS */
 4130  030144  041445  051117  020122  TMS35:  .ASCII  /*CORR */
       030152     043
 4131  030153     045  051103  020103  TMS36:  .ASCII  /*CRC */
       030160     043
 4132  030161     045  040523  020103  TMS37:  .ASCII  /*SAC */
       030166     043
 4133  030167     045  041506  020123  TMS38:  .ASCII  /*FCS */
       030174     043
 4134  030175     045  041501  046103  TMS39:  .ASCII  /*ACCL */
       030202  021440
 4135
 4136                                          .EVEN
 4137                                          ;WRITE BUFFER
 4138
 4139  030204  000100                  WDATA:
 4141  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
  (1)  030240  177777                           1
  (1)  030242  177777                          -1
```

```
  (1)  030244  177777                         -1
  (1)  030246  177777                         -1
  (1)  030250  177777                         -1
  (1)  030252  177777                         -1
  (1)  030254  177777                         -1
  (1)  030256  177777                         -1
  (1)  030260  177777                         -1
  (1)  030262  177777                         -1
  (1)  030264  177777                         -1
  (1)  030266  177777                         -1
  (1)  030270  177777                         -1
  (1)  030272  177777                         -1
  (1)  030274  177777                         -1
  (1)  030276  177777                         -1
  (1)  030300  177777                         -1
  (1)  030302  177777                         -1
  (1)  030304  177777                         -1
  (1)  030306  177777                         -1
  (1)  030310  177777                         -1
  (1)  030312  177777                         -1
  (1)  030314  177777                         -1
  (1)  030316  177777                         -1
  (1)  030320  177777                         -1
  (1)  030322  177777                         -1
  (1)  030324  177777                         -1
  (1)  030326  177777                         -1
  (1)  030330  177777                         -1
  (1)  030332  177777                         -1
  (1)  030334  177777                         -1
  (1)  030336  177777                         -1
  (1)  030340  177777                         -1
  (1)  030342  177777                         -1
  (1)  030344  177777                         -1
  (1)  030346  177777                         -1
  (1)  030350  177777                         -1
  (1)  030352  177777                         -1
  (1)  030354  177777                         -1
  (1)  030356  177777                         -1
  (1)  030360  177777                         -1
  (1)  030362  177777                         -1
  (1)  030364  177777                         -1
  (1)  030366  177777                         -1
  (1)  030370  177777                         -1
  (1)  030372  177777                         -1
  (1)  030374  177777                         -1
  (1)  030376  177777                         -1
  (1)  030400  177777                         -1
  (1)  030402  177777                         -1
 4142
 4143
 4144                                         ;READ BUFFER
 4145
 4146  030404  000100             RDATA:
 4148  030404  000000                         0
  (1)  030406  000000                         0
  (1)  030410  000000                         0
```

C10


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

```
  (1)  030572  000000                          0
  (1)  030574  000000                          0
  (1)  030576  000000                          0
  (1)  030600  000000                          0
  (1)  030602  000000                          0
4149
4150                                           ;WRAP AROUND MESSAGES******
4151
4152   030604  042045  052101  020101  WMSG27: .ASCII  /%DATA PAT:%/
       030612  040520  035124     043
4153   030617     045  047505  020122  WMSG31: .ASCII  /%EOR CLEAR DID NOT CLEAR GO%%/
       030624  046103  040505  020122
       030632  044504  020104  047516
       030640  020124  046103  040505
       030646  020122  047507  021445
4154
4155                                           .EVEN
4156   030654  000000                  PRE:    0
4159   030656  000000                          0
  (1)  030660  000000                          0
  (1)  030662  000000                          0
  (1)  030664  000000                          0
  (1)  030666  000000                          0
  (1)  030670  000000                          0
  (1)  030672  000000                          0
  (1)  030674  000000                          0
  (1)  030676  000000                          0
  (1)  030700  000000                          0
  (1)  030702  000000                          0
  (1)  030704  000000                          0
  (1)  030706  000000                          0
  (1)  030710  000000                          0
  (1)  030712  000000                          0
  (1)  030714  000000                          0
  (1)  030716  000000                          0
  (1)  030720  000000                          0
  (1)  030722  000000                          0
  (1)  030724  000000                          0
  (1)  030726  000000                          0
  (1)  030730  000000                          0
  (1)  030732  000000                          0
  (1)  030734  000000                          0
  (1)  030736  000000                          0
  (1)  030740  000000                          0
  (1)  030742  000000                          0
  (1)  030744  000000                          0
  (1)  030746  000000                          0
  (1)  030750  000000                          0
  (1)  030752  000000                          0
  (1)  030754  000000                          0
  (1)  030756  000000                          0
  (1)  030760  000000                          0
  (1)  030762  000000                          0
  (1)  030764  000000                          0
  (1)  030766  000000                          0
  (1)  030770  000000                          0
```

```
   (1)  030772  000000                          0
   (1)  030774  000000                          0
  4160  030776  000000                POST:    0
  4163  031000  000000                          0
   (1)  031002  000000                          0
   (1)  031004  000000                          0
   (1)  031006  000000                          0
   (1)  031010  000000                          0
   (1)  031012  000000                          0
   (1)  031014  000000                          0
   (1)  031016  000000                          0
   (1)  031020  000000                          0
   (1)  031022  000000                          0
   (1)  031024  0  000                          0
   (1)  031026  000000                          0
   (1)  031030  000000                          0
   (1)  031032  000000                          0
   (1)  031034  000000                          0
   (1)  031036  000000                          0
   (1)  031040  000000                          0
   (1)  031042  000000                          0
   (1)  031044  000000                          0
   (1)  031046  000000                          0
   (1)  031050  000000                          0
   (1)  031052  000000                          0
   (1)  031054  000000                          0
   (1)  031056  000000                          0
   (1)  031060  000000                          0
   (1)  031062  000000                          0
   (1)  031064  000000                          0
   (1)  031066  000000                          0
   (1)  031070  000000                          0
   (1)  031072  000000                          0
   (1)  031074  000000                          0
   (1)  031076  000000                          0
   (1)  031100  000000                          0
   (1)  031102  000000                          0
   (1)  031104  000000                          0
   (1)  031106  000000                          0
   (1)  031110  000000                          0
   (1)  031112  000000                          0
   (1)  031114  000000                          0
   (1)  031116  000000                          0
  4164  031120  000000                WBUFF:    0
  4165        031532                          .+410
  4166  031532  000000                RBUFF:    0
  4167
  4168        000001                          .END
```

K L0

LT4A	003716	1875	1889#	1927	1933
LT4B	003760	1894	1898#		
LT4C	003766	1900#			
LT4D	004120	1896	1924#	1935	
LT4ERG	004146	1929	1931#		
LT4ER1	004130	1897	1928#		
LT4ER2	004140	1899	1930#		
LT4G	003646	1878#	1925		
LT4GO	003636	1876#			
LT4X	004176	1873	1886	1923	1936#
LT40	012152	1540	2866#		
LT40IT	012166	1541	2866	2868#	
LT40X	012262	2877	2883#		
LT40XX	012266	2884#			
LT41	012272	1542	2889#		
LT41IT	012306	1543	2889	2891#	
LT41X	012502	2914	2918	2920#	
LT42	012534	1544	2936#		
LT42A	012636	2950#	2953		
LT42B	012652	2951	2954#		
LT42B1	012700	2955	2960#		
LT42B2	012720	2959	2964#		
LT42C	012764	2971#	2974		
LT42D	013000	2972	2975#		
LT42E	013030	2976	2981#		
LT42IT	012572	1545	2942	2943#	
LT42X	013050	2980	2984	2986#	
LT43	013064	1546	2992#		
LT43C	013174	3008#	3011		
LT43D	013210	3009	3012#		
LT43E	013242	3013	3018#		
LT43F	013274	3022	3025#		
LT43IT	013122	1547	2996	2998#	
LT43X	013314	3017	3028	3030#	
LT43XX	013324	2993	3032#		
LT44	013330	1548	3036#		
LT44A	013414	3046#	3049		
LT44A1	013426	3047	3050#		
LT44B	013436	3052#	3055		
LT44C	013452	3053	3056#		
LT44D	013500	3057	3061#		
LT44E	013502	3062#			
LT44F	013542	3063	3070#		
LT44IT	013356	1549	3039	3040#	
LT44X	013562	3069	3073	3075#	
LT44XX	013572	3077#			
LT45	013576	1550	3082#		
LT45A	013704	3098#	3101		
LT45B	013720	3099	3102#		
LT45D	013750	3103	3107#		
LT45E	013662	3092#	3095		
LT45E1	013674	3093	3096#		
LT45F	014002	3111	3114#		
LT45IT	013624	1551	3085	3086#	
LT45Z	014026	3118	3120#		
LT45XX	014036	3122#			

```
LT4b    014042          1552    3127#
LT46A   014124          3136#   3137
LT46B   014162          3144#   3147
LT46C   014176          3145    3148#
LT46D   014230          3149    3154#
LT46IT  014070          1553    3130    3131#
LT46X   014270          3153    3161    3163#
LT46XX  014300          3165#
LT47    014304          1554    3169#
LT47A   014374          3179#   3182
LT47B   014406          3180    3183#
LT47C   014412          3184#   3187
LT47IT  014320          1555    3170    3171#
LT47X   014464          3191    3195#
LT5     004202          1486    1941#
LT5A    004222          1945#   1953    1967
LT5B    004242          1950#   1969
LT5C    004252          1951    1954#
LT5D    004262          1956#   1965    1971
LT5E    004304          1960    1962#   1973
LT5ER1  004316          1949    1966#
LT5ER2  004340          1961    1970#
LT5IT   004210          1487    1942#
LT5X    004362          1963    1974#
LT50    014474          1556    3202#
LT50IT  014510          1557    3203    3204#
LT50X   014634          3221    3223    3225#
LT51    014650          1558    3234#
LT51IT  014664          1559    3235    3236#
LT51X   015010          3253    3255    3257#
LT6     004372          1488    1980#
LT6A    004410          1984#   1995
LT6A1   004406          1983#   1993
LT6B    004414          1985#   1997
LT6D    004434          1990#   1999
LT6ER1  004454          1989    1996#
LT6IT   004374          1489    1981#
LT6X    004476          1991    2000#
LT7     004506          1490    2006#
LT7A    004524          2010#   2020    2022
LT7B    004544          2015#   2024
LT7C    004514          2008#   2018
LT7ER1  004564          2014    2021#
LT7IT   004510          1491    2007#
LT7X    004606          2016    2025#
MMSG0   027122          3642    4079#
MMSG1   027147          2161    4080#
MMSG2   027245          2184    4081#
MMSG3   027266          2206    4082#
MMSG4   027324          2232    4083#
MMSG5   027437          4084#
MR      000534          1352#   1945*   1946    1956*   1957    2065*   2105*   2289*   2345*   2372*   2385*   2387    2390
                        2426*   2455*   2461*   2462*   2518*   2529*   2543*   2543*   2576*   2600*   2626    2628    2655
                        2641    2652    2755*   2798*   2850*   2875*   2894*   2947*   2968*   3002*   3018    3050*   3096*
                        3138    3140    3142*   3176*   3184*   3185*   3210*   3242*   3276    3279*   3280*   3586
MSG1    020762          1588    1593*   3961#
```

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

MSG68	023730	1915	4030#											
MSG69	024002	1911	4031#											
MSG7	021454	3330	3971#											
MSG8	021472	1878	3972#											
MSG8A	021533	1876	3973#											
MSG9	021662	1932	3975#											
MSLT1	024012	1745	4034#											
MSLT10	024501	2033	4041#											
MSLT11	024557	2060	4042#											
MSLT12	024624	2101	4043#											
MSLT13	024675	2133	4044#											
MSLT14	024742	2160	4045#											
MSLT15	025051	2183	4046#											
MSLT16	025141	2205	4047#											
MSLT17	025230	2231	4048#											
MSLT2	024066	1798	4035#											
MSLT20	025311	2256	4049#											
MSLT21	025370	2285	4050#											
MSLT22	025424	2315	4051#											
MSLT23	025461	2339	4052#											
MSLT24	025526	2364	4053#											
MSLT25	025566	2422	4054#											
MSLT26	025622	2447	4055#											
MSLT27	025656	2491	4056#											
MSLT3	024145	1834	4036#											
MSLT30	025712	2513	4057#											
MSLT31	025750	2567	4058#											
MSLT32	026004	2715	4059#											
MSLT33	026040	2752	4060#											
MSLT34	026074	2770	4061#											
MSLT35	026130	2793	4062#											
MSLT36	026175	2818	4063#											
MSLT37	026242	2839	4064#											
MSLT4	024232	1931	4037#											
MSLT40	026310	2867	4065#											
MSLT41	026355	2890	4066#											
MSLT42	026424	2941	4067#											
MSLT43	026460	2997	4068#											
MSLT44	026514	3038	4069#											
MSLT45	026574	3084	4070#											
MSLT46	026656	3129	4071#											
MSLT47	025712	3169	4072#											
MSLT5	024314	1941	4038#											
MSLT50	026762	3202	4073#											
MSLT51	027042	3234	4074#											
MSLT6	024363	1981	4039#											
MSLT7	024432	2008	4040#											
MS57A	023356	1627	4019#											
MTINT	017246	1334	1613	3654#										
NONSTD	001020	1451#												
NOST	016670	1636	3570#											
NRZOF	000656	1402#												
NXTDRV	002336	1700#												
NXTSLV	002410	1700#	1726											
OCTP	020172	1597	1606	1640	1649	1658	1668	1730	3354	3353	3507	3511	3513	3830#
OCTPE	020162	3342	3346	3371	3373	3375	3377	3379	3381	3383	3385	3387	3389	3828#

```
                                3946
OCTPE1   020176                 3829     38310
OCTPG    020350                 3840     3843     3850     3854     3859     3864     3866     38710
OCTPG0   020366                 3872     3874     38770
OCTPG1   020372                 3835     38780
OCTP0    020216                 3832     3834     38570
OCTP1    020236                 3838     38420
OCTP2    020244                 3841     38440
OCTP3    020334                 3836     38670
OFL      020412                 3828*    3830*    3833     3873     3877*    38830
PATRN    000776                 14420
PCNTR    001014                 14490    1685*    1729     1741*    3550
PEXFL    000732                 14240    3490*    3502     3528*
PFLG     000662                 14040    3311*    3332*    3357     3400*    3434*    3438     3476*    3481     3485*    3492*    3518*    3523
POST     030776                 41600
PRE      030654                 41560
PREFL    000764                 14370
PSW      000566                 13710    2137*    2141*    3666*
RBUFF    031532                 41660
RCDP     001004                 14450
RDAD     000756                 14340
RDATA    030404                 41460
RDRVF    001002                 14440    1682*    3269*
REGP     015534                 3347     33670    3433     3475
REGS     000606                 13790    1596     1598     1615
RTRN     000664                 14050    2135*    3655
SAV1     000700                 14110
SAV2     000702                 14120
SAV3     000704                 14130
SCF      000726                 14220
SCHN     001450                 15820
SCOLP    000706                 14140    1787*    1825*    1849*    1854*    1955*    1967*    1971*    1997*    2022*    2050*    2080*    2086*
                                2092*    2116*    2120*    2124*    2143*    2168*    2191*    2213*    2239*    2257*    2286*    2316*    2340*
                                2365*    2432*    2472*    2490*    2514*    2568*    2594*    2716*    2755*    2778*    2785*    2803*    2819*
                                2823*    2830*    2846*    2857*    2866*    2889*    2942*    2996*    3039*    3085*    3130*    3170*    3203*
                                3235*    3443*    3542
SCOPE    016550                 2540     3363     3444     3529     35340
SERFL    000766                 14380
SKAT     001012                 14480    1580*    1740*    1748     1820
SCAF     000716                 14180
SCVN     000660                 14030    1648     1650     1700*    1868     1869     2218     3586     3609
SCVTYP   001010                 14470    1667     1669     1674     1694*    1913     2626
SN       000540                 13540    1920
SNPG     020574                 3918     3921     3927     3929     39310
SNPT     020516                 1921     39120
SSCF     000720                 14190
START    001334                 1325     15680
STATC    001006                 14460    1657     1659     3665
STATF    001000                 14430
STATIC   015024                 2937     2995     3037     3087     3128     32650
STFLG    000734                 14250    1696*    1712     1717*    1750     1822     2157     2180     2202     2228     3243
STSCD    002566                 1699     17170
STW016   006020                 1676*    2209     22180
STW017   006132                 1677*    2235     22440
ST0      001664                 16180    1622
ST1      001706                 16240    1626
```

```
STZ      002226          1329    1675    1681#
SWR      000570          1372#   1573    1577*   1581*   1697    1708    1714    1737    1752    1772    1874    1922    2155
                         2178    2200    2226    3289    3297    3312    3348    3354    3359    3425    3435    3440    3467
                         3478    3483    3493    3512    3520    3525    3535    3546    3671    3673*   3681*   3940    3945
                         3949
SWREG    000176          1321#   1577    1581    3671    3681    3940
TADX     001326          1560#   3572
TC       000542          1355#   1700*   1891*   1985*   1986    2067*   2107*   2288*   2343*   2368*   2424*   2452*   2516*
                         2526*   2532    2575*   2599*   2726*   2773*   2774*   2797*   2800    2806*   2807    2821    2828
                         2842*   2843    2849*   2853    2872*   2892*   3174*   3179    3205*   3206*   3237*   3238*   3388
                         3586*   3609*   3624*
TEMP1    000670          1407#   1763    1885    2259*   2263    2276*   3714*   3723    3747*
TEMP2    000672          1408#   3301*
TEMP3    000674          1409#   1746*   1758    1774    1777    1868*   1880    1893    3451*   3455*   3458*   3473    3644
                         3903    3906
TEND     002622          1560    1719    1725#   1917
TENDX    002720          1736    1738    1740#
TEX      020160          3790    3823#
TIB      000612          1384#   3717    3721    3726    3731    3741    3744    3751*   3752    3780*   3781*   3782
TIMER    010642          2584    2607    2642    2645#   2651    2653
TIMERO   010666          2652#
TIMER1   010616          2641#   2644
TIMOK    010734          2590    2613    2669#
TIMON    010550          2580    2603    2623#
TIMOVF   010700          2649    2656#
TINER    017676          3743    3746    3757    3760    3769#
TKB      000574          1374#   3660    3780    3809
TKS      000572          1373#   1689*   3777*   3778    3807
TLAST    001330          1561#   1718    3574*   3575
TMS1     027513          4091#
TMS10    027601          2499    4100#
TMS11    027607          4101#
TMS12    027615          4102#
TMS13    027623          4103#
TMS14    027631          2760    4104#
TMS15    027637          4105#
TMS16    027645          4106#
TMS17    027653          2267    4107#
TMS18    027661          4108#
TMS19    027667          2296    4109#
TMS2     027521          4092#
TMS20    027675          2325    4110#
TMS21    027704          2350    4111#
TMS22    027712          2401    4112#
TMS23    027721          3104    4113#
TMS24    027727          4114#
TMS25    027735          3150    4115#
TMS26    027743          3014    4116#
TMS27    027751          4117#
TMS28    027757          2473    2911    4118#
TMS29    027765          4119#
TMS3     027527          2880    2901    4094#
TMS30    027772          2906    4120#
TMS31    030000          2433    4121#
TMS32    030006          2512    4122#
TMS33    030014          4123#
```

TMS33A	030022	2569	41240											
TMS33B	030036	2594	41250											
TMS33C	030051	2684	41260											
TMS33D	030073	2695	41270											
TMS33E	030116	2660	41280											
TMS34	030136	2733	41290											
TMS35	030144	3058	41300											
TMS36	030153	2977	41310											
TMS37	030161	2792	41320											
TMS38	030167	2819	41330											
TMS39	030175	2840	41340											
TMS4	027534	40940												
TMS5	027540	40950												
TMS6	027551	2769	40960											
TMS7	027557	40970												
TMS8	027565	40980												
TMS9	027573	40990												
TOB	000610	13830	1622	3782*	3788*	3789	3791	3795*	3798*	3802*	3822	3867*	3879*	3890*
		3893	3898*	3931*	3933*									
TOG	020060	3785	3793	3796	3799	3803	38070	3814	3819	3821	3868	3880	3934	
TPB	000600	13760	3822*	3895*	3897*									
TPS	000576	13750	3820	3891										
TREF	000730	14230												
TR00	000622	13880	1801											
TR01	000624	13890												
TR02	000626	13900												
TR03	000630	13910												
TR04	000632	13920												
TR05	000634	13930												
TR06	000636	13940												
TR07	000640	13950												
TR10	000642	13960												
TR11	000644	13970												
TR12	000646	13980												
TR13	000650	13990												
TR14	000652	14000												
TR15	000654	14010												
TSCD	002246	1581	1592	16890	1716	1742								
TSCDA	002466	1700	17010											
TSCDO	002474	17020	1713											
TSCD1	002502	17030	1724											
TSCD2	002530	17080	1790	1829	1863	1936	1975	2001	2027	2054	2095	2127	2146	2172
		2195	2217	2243	2280	2310	2334	2360	2417	2442	2483	2507	2561	2617
		2742	2764	2787	2813	2834	2861	2884	2927	2963	2988	3032	3077	3122
		3165	3196	3227	3259	3268	3565							
TSCD3	002546	1709	17120											
TSTTBL	001056	14760	1701	1722	3572*	3575*								
TTIN	017716	3716	37770											
TTINT	017256	1315	36600											
TTOUT	017760	1586	1589	1595	1604	1628	1638	1647	1656	1666	1728	1757	1829	1901
		1912	1916	1919	2537	2539	2657	2659	2661	2681	2683	2685	2692	2694
		2696	2962	3294	3296	3317	3320	3331	3338	3340	3344	3351	3360	3430
		3432	3470	3474	3498	5501	5505	5509	3515	3563	3641	3643	5616	5729
		3738	3770	37880	3794	3804	5944	3948						
TTR	017436	1602	1611	1633	1645	1654	1663	1673	1762	1884	5648	57120	5953	
T24H	000740	14270	2400*	2404*	3335									

```
UDES    000772  1440#  2936*  2994*  3036*  3082*  3127*  3595  3596*  3600  3601*  3606  3610  3624
                3633*
VECT    000604  1378#  1605   1607   1612
WAM     000744  1429#
WBUFF   031120  4164#
WC      000512  1343#  2261*  2371*  2453*  2944*  2965*  3003*  3041*  3087*  3132*  3172*  3207*  3239*
                3372
WCDP0   001044  1467#
WCDP2   001032  1462#
WCS1    001022  1455#
WCS2    001024  1456#
WDATA   030204  2369   2946   2967   3005   3043   3089   3134   3175   3209   3241   4139#
WDS     001026  1457#
WER     001030  1458#
WMSG27  030604  3337   4152#
WMSG31  030617  3295   4153#
WPGFL   000774  1441#  1695*  1711*
WTAD    000752  1432#
W2FLG   000760  1435#
$CNTRL  001332  1562#  3686*  3690*  3809*  3810*  3811  3813*  3815  3817*  3818
$DONE   002630  1700   1727#
$ENDAD  002666  1511   1734#
$MNEW   027503  3947   4090#
$MSWR   027473  3943   4089#
$SVPC   000764  1311#
.       031534  1310#  1311#  1314#  1320#  1324#  1328#  1333#  1339#  2644#  2651#  4165#
.RESTO  020740  3677   3954   3957#
.SAVE   020716  3674   3942   3956#
```

```
$CATCH   1228#   1310
$CHAIN   1228#   1581
$CHNMO   1228#   1700
$RESTO   1228#   3957
$SAVE    1228#   3956
.$ACT1   1228#   1311
.$EOP    1228#   1734


. ABS.  031534     000


 ERRORS DETECTED:  0

 CZTEAE,CZTEAE/CRF=CZTEAE.SML/ML,CZTEAE.P11
 RUN-TIME:  4  8  1 SECONDS
 RUN-TIME RATIO: 20/14=1.4
 CORE USED:  14K  (28 PAGES)
```

CZTEAE.P11	06-APR....B1	CZTEAE.P11	06-APR....B5	CZTEAE.P11	06-APR....B9
CZTEAE.P11	06-APR....C1	CZTEAE.P11	06-APR....C5	CZTEAE.P11	06-APR....C9
CZTEAE.P11	06-APR....D1	CZTEAE.P11	06-APR....D5	CZTEAE.P11	06-APR....D9
CZTEAE.P11	06-APR....E1	CZTEAE.P11	06-APR....E5	CZTEAE.P11	06-APR....E9
CZTEAE.P11	06-APR....F1	CZTEAE.P11	06-APR....F5	CZTEAE.P11	06-APR....F9
CZTEAE.P11	06-APR....G1	CZTEAE.P11	06-APR....G5	CZTEAE.P11	06-APR....G9
CZTEAE.P11	06-APR....H1	CZTEAE.P11	06-APR....H5	CZTEAE.P11	06-APR....H9
CZTEAE.P11	06-APR....I1	CZTEAE.P11	06-APR....I5	CZTEAE.P11	06-APR....I9
CZTEAE.P11	06-APR....J1	CZTEAE.P11	06-APR....J5	CZTEAE.P11	06-APR....J9
CZTEAE.P11	06-APR....K1	CZTEAE.P11	06-APR....K5	CZTEAE.P11	06-APR....K9
CZTEAE.P11	06-APR....L1	CZTEAE.P11	06-APR....L5	CZTEAE.P11	06-APR....L9
CZTEAE.P11	06-APR....M1	CZTEAE.P11	06-APR....M5	CZTEAE.P11	06-APR....M9
CZTEAE.P11	06-APR....N1	CZTEAE.P11	06-APR....N5	CZTEAE.P11	06-APR....N9

CZTEAE.P11	06-APR....B2	CZTEAE.P11	06-APR....B6	CZTEAE.P11	06-APR....B10
CZTEAE.P11	06-APR....C2	CZTEAE.P11	06-APR....C6	CZTEAE.P11	06-APR....C10
CZTEAE.P11	06-APR....D2	CZTEAE.P11	06-APR....D6	CZTEAE.P11	06-APR....D10
CZTEAE.P11	06-APR....E2	CZTEAE.P11	06-APR....E6	CZTEAE.P11	06-APR....E10
CZTEAE.P11	06-APR....F2	CZTEAE.P11	06-APR....F6	CZTEAE.P11	06-APR....F10
CZTEAE.P11	06-APR....G2	CZTEAE.P11	06-APR....G6	CZTEAE.P11	06-APR....G10
CZTEAE.P11	06-APR....H2	CZTEAE.P11	06-APR....H6	CZTEAE.P11	06-APR....H10
CZTEAE.P11	06-APR....I2	CZTEAE.P11	06-APR....I6	CZTEAE.P11	06-APR....I10
CZTEAE.P11	06-APR....J2	CZTEAE.P11	06-APR....J6	CZTEAE.P11	06-APR....J10
CZTEAE.P11	06-APR....K2	CZTEAE.P11	06-APR....K6	CZTEAE.P11	06-APR....K10
CZTEAE.P11	06-APR....L2	CZTEAE.P11	06-APR....L6	CZTEAE.P11	06-APR....L10
CZTEAE.P11	06-APR....M2	CZTEAE.P11	06-APR....M6	CZTEAE.P11	06-APR....M10
CZTEAE.P11	06-APR....N2	CZTEAE.P11	06-APR....N6	CZTEAE.P11	06-APR....N10

CZTEAE.P11	06-APR....B3	CZTEAE.P11	06-APR....B7	CZTEAE.P11	06-APR....B11
CZTEAE.P11	06-APR....C3	CZTEAE.P11	06-APR....C7	CZTEAE.P11	06-APR....C11
CZTEAE.P11	06-APR....D3	CZTEAE.P11	06-APR....D7	CZTEAE.P11	06-APR....D11
CZTEAE.P11	06-APR....E3	CZTEAE.P11	06-APR....E7	CZTEAE.P11	06-APR....E11
CZTEAE.P11	06-APR....F3	CZTEAE.P11	06-APR....F7	CZTEAE.P11	06-APR....F11
CZTEAE.P11	06-APR....G3	CZTEAE.P11	06-APR....G7		
CZTEAE.P11	06-APR....H3	CZTEAE.P11	06-APR....H7		
CZTEAE.P11	06-APR....I3	CZTEAE.P11	06-APR....I7		
CZTEAE.P11	06-APR....J3	CZTEAE.P11	06-APR....J7		
CZTEAE.P11	06-APR....K3	CZTEAE.P11	06-APR....K7		
CZTEAE.P11	06-APR....L3	CZTEAE.P11	06-APR....L7		
CZTEAE.P11	06-APR....M3	CZTEAE.P11	06-APR....M7		
CZTEAE.P11	06-APR....N3	CZTEAE.P11	06-APR....N7		

CZTEAE.P11	06-APR....B4	CZTEAE.P11	06-APR....B8
CZTEAE.P11	06-APR....C4	CZTEAE.P11	06-APR....C8
CZTEAE.P11	06-APR....D4	CZTEAE.P11	06-APR....D8
CZTEAE.P11	06-APR....E4	CZTEAE.P11	06-APR....E8
CZTEAE.P11	06-APR....F4	CZTEAE.P11	06-APR....F8
CZTEAE.P11	06-APR....G4	CZTEAE.P11	06-APR....G8
CZTEAE.P11	06-APR....H4	CZTEAE.P11	06-APR....H8
CZTEAE.P11	06-APR....I4	CZTEAE.P11	06-APR....I8
CZTEAE.P11	06-APR....J4	CZTEAE.P11	06-APR....J8
CZTEAE.P11	06-APR....K4	CZTEAE.P11	06-APR....K8
CZTEAE.P11	06-APR....L4	CZTEAE.P11	06-APR....L8
CZTEAE.P11	06-APR....M4	CZTEAE.P11	06-APR....M8
CZTEAE.P11	06-APR....N4	CZTEAE.P11	06-APR....N8