Model 110

1/4" Cartridge Tape Coupler

600-260-00 A

REVISION HISTORY		
ECO #	DATE	DESCRIPTION
0328	6/28/84	New ZETACO Cover

TABLE OF CONTENTS

1.0 INTRODUCTION

The Custom Systems Model 110 Cartridge Tape Coupler (CTC) interfaces any Data General Nova* or Eclipse* Minicomputer to a Kennedy Model 6450 or 6455 1/4" Cartridge Tape System. It offers the capability of storing up to 20Mb of data on a 600 foot cartridge utilizing a 4K word block size. The CTC uses the Data Channel to transfer up to 4K word blocks of data to or from the Tape System.

The Model 110 fully supports the Data General 9 Track Tape Subsystem (6026) as well as all the disk back-up functions: DUMP, F-DUMP, BURST, BACK-UP, etc.

Data is written on the tape in a true "serpentine" fashion using 4 tracks. This, in effect, allows a 600 foot cartridge to look like a 2400 foot continuous reel of tape. Track management and all command functions are controlled by an on-board Microprocessor for complete transparency to the operating system.

1.1 FEATURES

- Microprocessor based design allows extensive self-test with led indicator for visual fault isolation.
- Memory Addressing to 32K words.
- Transfer up to 4K word blocks.
- Supports two (2) Drives.
- Fully transparent to Data General operating Systems: RDOS, AOS, etc.

*Trademark of Data General

1.1 FEATURES (continued)

- Emulates 6026 9 Track PE Tape Subsystem.
- Low power design requires only +5V.

2.0 SPECIFICATIONS

2.1 PERFORMANCE

Recording Format	- 4 Track - Serpentine (See Appendix B)
Transfer Rate	- 24K bytes/sec, 192K bits/sec
Tape Speed	- Read/Write - 30 IPS - Rewind - 90 IPS
Recording Density	- 6400 BPI
Error Rates	- Soft-1 in 10^{10} Bits - Hard-1 in 10^{11} Bits
Tape Length	- 300, 450 or 600 feet
Start/Stop Time	- Read/Write - 25 msec - Rewind/Search - 75 msec
Start/Stop Displacement	- Read/Write - .38 in. - Rewind/Search - 3.38 in.

2.2 INTERFACE

Cabling	- Single 34 Pin Ribbon Cable
Electrical	- TTL (low true) - True = Low = .4V Max - False = High = 2.4V Min
Receiver	- Schmidt Trigger
Handshake	- Standard Kennedy Pico Bus*

*Trademark of Kennedy

2.3 POWER REQUIREMENTS

Drive	- +5V - DC @ 2.5A (avg.) - 5A (pk) - +24V - DC @ 1.5A (avg.) - 3A(pk)
Controller	- +5V @ 2.5A (max)
Power Supply	- 100W (max) Output

2.4 PHYSICAL

Dimensions	- Drive	- 4.5"H x 8.5"W x 17.5"D
	- Controller	- 0.5"H x 15.0"W x 15.0"D
	- Power Supply	- N/A
Weight	- Drive	- 8 lbs./17.6 kg
	- Controller	- 5 lbs./11 kg
	- Power Supply	- N/A

2.5 ENVIRONMENTAL

Operating Temp	- Drive	- 0^{o} to 55^{o}C
	- Controller	- 0^{o} to 55^{o}C
	- Power Supply	- N/A
Humidity	- Drive	- 10% to 90% (non-condensing)
	- Controller	- 10% to 90%
	- Power Supply	- N/A

3.0 INSTALLATION INSTRUCTIONS

3.1 INSPECTION

Upon arrival the package and board should be inspected for any obvious damage. If any damage is apparent contact the carrier and Custom Systems stating the nature and extend of the damage.

3.2 BOARD INSERTION

Since the CTC is a Data Channel Device its placement in the CPU or Expansion Chassis is critical. The closer to the CPU, the higher the priority guaranteeing minimal Data Channel latency.

Once a slot has been chosen, the board should be carefully guided into the slot with the edges of the board resting in the Card Guides on either side of the Chassis. When the board meets the edge connector, the lock tabs are used in conjunction with the two outside edges to provide additional leverage. Use equal pressure on both lock tabs and seat the board firmly into the backplane connectors.

3.3 PRIORITY JUMPERS

With the board installed in the proper slot, two priority signals are required to allow the CTC to communicate with the system. These signals are Data Channel Priority In - ($\overline{\text{DCHPIN}}$ - A94) and Interrupt Priority In - ($\overline{\text{INTPIN}}$ - A96). If any vacant slots exist between the CTC and the processor, priority jumpers must be installed to maintain priority continuity between controllers. To jumper across these slots, connect a jumper between A94 and A93 ($\overline{\text{DCHPOUT}}$) and between A96 and A95 ($\overline{\text{INTPOUT}}$) of the vacant slot(s).

3.4 CABLING

The CTC requires only one external cable to interface to the Kennedy Model 645X 1/4" Cartridge Tape Unit. This 34 Pin Ribbon Cable connects the Drive to either J1 or J2 on the handle end of the CTC board. If the Drive is plugged into J1 it appears to the operating system as Unit 0, plugging the Drive into J2 makes it appear as Unit 1.

CAUTION: When installing cables, carefully check to be sure the arrows on the connectors and cables match. If installed wrong the units will appear not ready.

3.5 CONFIGURATION

The only configuration of the CTC is determined by the operating system. If the CTC is being operated in an AOS environment cut J4, (near Location L8) otherwise leave it in.

NOTE: With J4 cut one recoverable parity error will be seen when crossing track boundries. This is considered a "soft error".

4.0 ADDRESSING

The Cartridge Tape Coupler responds to Device Code 22_8 (Primary) and 62_8 (Secondary). Jumper J1 (near Location C2) is used to select between the two Device Codes as follows:

J1 - Out = Device Code 22_8

J2 - In = Device Code 62_8

Two non-standard Device Codes are available for use in special applications. These non-standard Device Codes: 20_8 (Primary) 60_8 (Secondary) are set up by removing Jumper J2 (near Location E2) and following the procedure described above.

5.0 PROGRAMMING NOTES

5.1 FUNCTION COMMANDS

Functions

C (Clear) - Clears all error flags (except EOT/BOT) and Done and Busy flags. Resets the Command to Read and Select Unit 0. If a CLR is issued during a Command the operation is aborted and Done is not set.

S (Start) - Clear all errors (except illegal), Set Busy and Clear Done. Command that was issued by a DOA, will be executed.

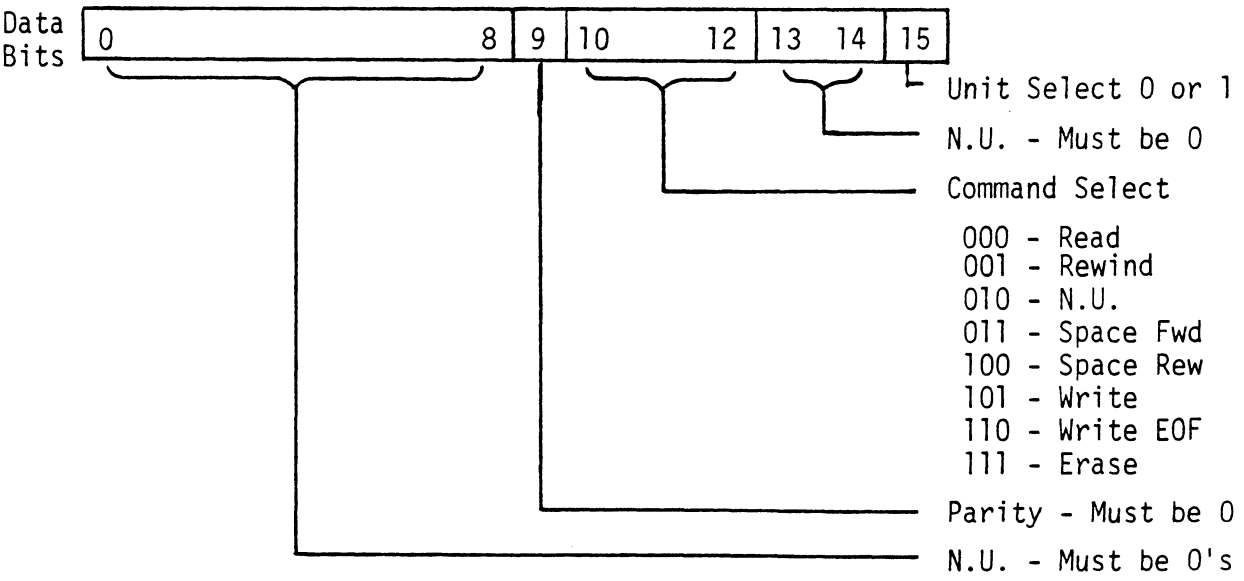
P (Pulse) - Not Used

Mask Bit 10

5.2 INSTRUCTIONS

1) DOA(F) AC, MTA - SEND COMMAND -

Before execution, Accumulator AC should contain:



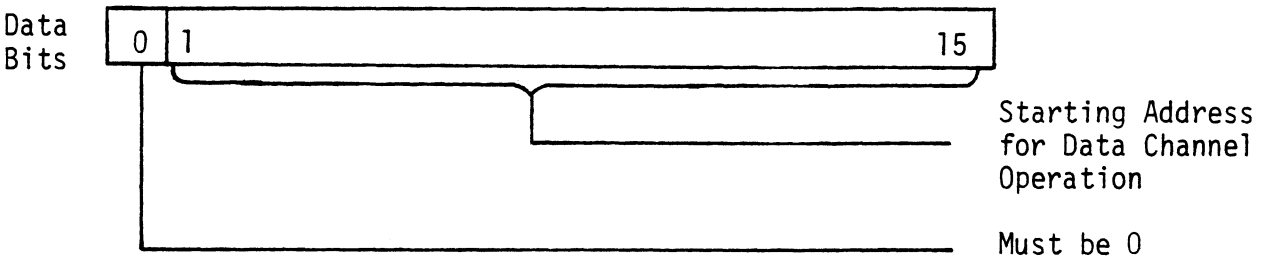
DOA(F) AC, MTA (continued)

NOTE: Command Default after a Clear or IORESET will be Read Operation, Unit 0.

- The controller will only access two (2) drives, Units 0 and 1. Only one command may be executed at a time with the exception of rewind.
- If only one Drive is attached it should be Unit 0.

2) DOB(F) AC, MTA - LOAD STARTING ADDRESS -

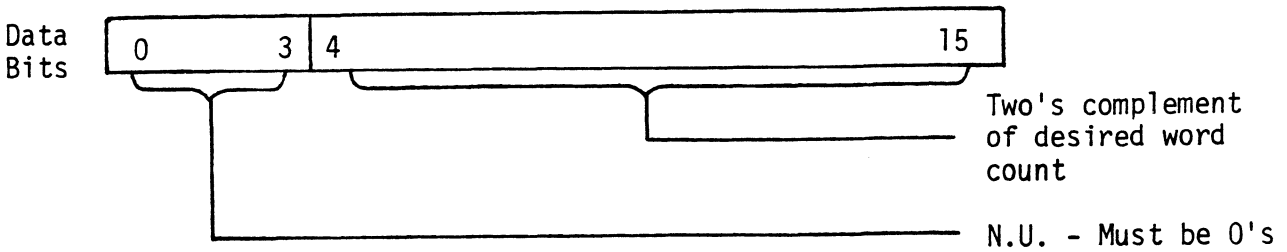
Before execution, Accumulator AC should contain:



NOTE: The contents of the accumulator is loaded into the Address Counter of the controller, which is then used as a base or Starting Address for subsequent Data Channel transfers.

3) DOC(F) AC, MTA - LOAD WORD COUNT -

Before execution, Accumulator AC should contain:



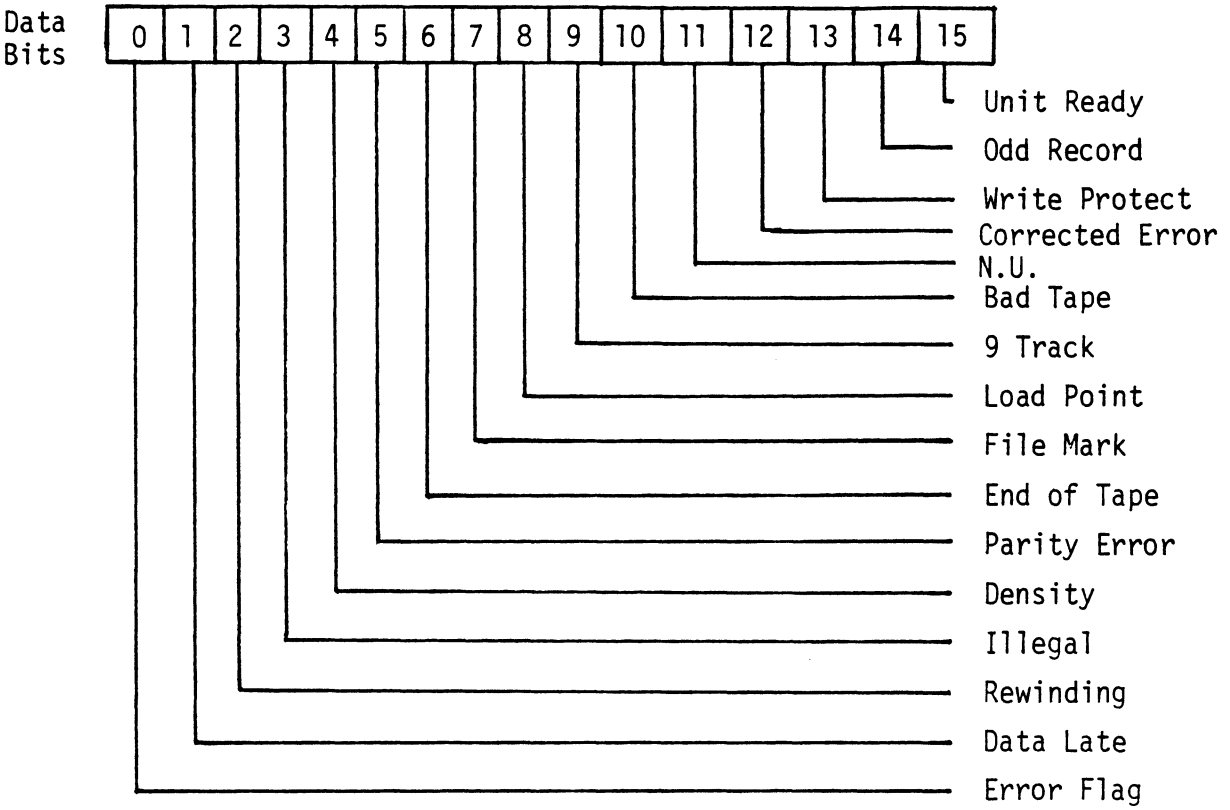
NOTE: Two's complement of the number of words to be transferred during the next Data Channel operation.

- If a space FWD/REW Command is issued the above accumulator should contain the two's complement of the number of records to be spaced.
- The maximum number of words to be transferred or records to be spaced is 4096.

5.2 INSTRUCTIONS (continued)

4) DIA(F) AC, MTA -READ CURRENT STATUS -

Following execution, Accumulator AC will contain:



DATA BIT	SIGNIFICANCE
0	Attention Flag - The controller detected a condition requiring service. Bit 1, 3, 5, 6, 7, 8, 10 or 14 is a one.
1	Data Late - Data Channel requests were not honored in time, resulting in lost data.
2	Rewinding - The Unit currently selected is rewinding.

4) DIA(F) AC, MTA (continued)

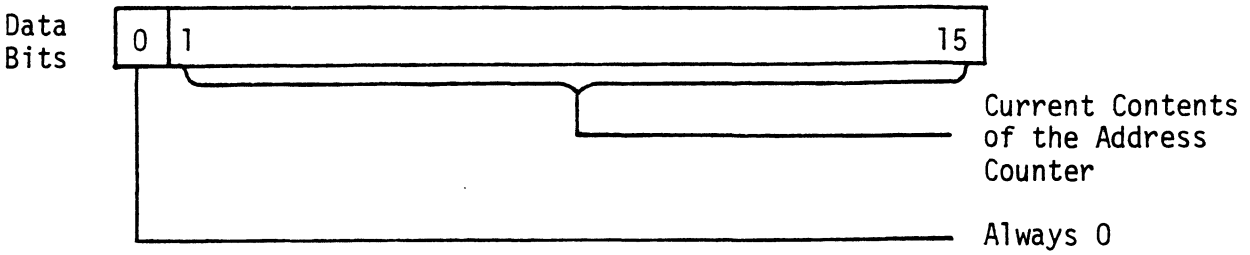
DATA BIT		SIGNIFICANCE
3	Illegal	- A start function was asserted under one of the following conditions: 1. A Write, Erase or Write FM issued with Write Protect on (Cartridge Lock in "Safe" position) 2. Space Reverse issued at Load Point 3. Space Forward, Write or Read issued with Unit at EOT 4. Unit was not ready NOTE: No tape motion will take place and Done will set
4	Density	- Always a one.
5	Parity Error	- A Parity Error was detected by the Drive on a Write or detected by the Coupler on a Read operation.
6	End of Tape	- The selected Unit is at or beyond EOT - Space Reverse or Rewind will clear this bit.
7	File Mark	- The controller has detected or written a File Mark.
8	Load Point	- The selected Unit is at Load Point (BOT).

4) DIA(F) AC, MTA (continued)

DATA BIT		SIGNIFICANCE
9	9 Track	- Not used, but will be a one for emulation.
10	Bad Tape	- One of the following conditions exist: 1. The Drive detected a CRC Error on the Tape 2. A faulty erase has occured 3. The Tape Cartridge is bad NOTE: A re-try may correct the above problem
11	Not Used	- Always 0
12	Corrected Error	- If this bit is a one after a Write Command the Parity Error Flag will also be set to a one and the Software should backspace and rewrite the record.
13	Write Protect	- The Tape Cartridge in Unit selected is Write Protected.
14	Odd Record	- An Odd Number of characters were Read within the previous record.
15	Unit Ready	- The following conditions must be satisfied before the Unit is ready: 1. Tape Cartridge inserted and properly loaded 2. The Drive/Formatter are functional 3. The Controller is not busy 4. The selected unit is not rewinding 5. Controller Self Test complete

5) DIB(F) AC, MTA - READ CURRENT ADDRESS -

Following execution, Accumulator AC will contain:



Read or Write Operation - Contains the Memory Address which will be involved in the next Data Channel operation. This counter is incremented by one after every Data Channel transfer.

Spacing Forward/Reverse - The Address Counter becomes a Record Counter during Space Forward or Reverse operations. The difference between the contents of the Counter before and after the Space Command will indicate the number of records spaced over.

6.0 COMMAND SUMMARY

6.1 READ

The program specifies a two's complement word count and a Starting Address. When Start sets Busy, tape motion begins. Data is then transferred a byte at a time to the controller, which then transfers data a word (2 bytes) at a time to the Mini's Memory. This operation continues until it reaches the end of the record or the Word Counter overflowed. If the record encountered is a filemark, Done sets and no data is transferred.

Possible Errors -

1. Unit not ready
2. Parity Error
3. Bad Tape
4. Data Late

6.2 REWIND

Start does not set Busy. Selected Unit rewinds at 90 IPS and will not be ready until it reaches Load Point. The operation takes from 1 ms to 90 sec. The other Unit is available for commands while this Unit is rewinding. Done will <u>not</u> set when the operation is complete.

6.3 SPACE FORWARD

The program specifies the two's complement of the number of records to be spaced forward. Start sets Busy and tape motion begins. The tape motion continues until one of the following occur:

1. End of tape was reached
2. A file mark was detected
3. The specified number of records were spaced over

Done will then set.

6.4 SPACE REVERSE

The program specifies the two's complement of the number of records to back-up. Start sets Busy and tape motion will begin unless the Drive is at Load Point, in which case Done and Illegal will set. If not at Load Point, tape motion will continue until one of the following occur:

1. Load Point was reached
2. A file mark was detected
3. The specified number of records were spaced over

Done will then set.

6.5 WRITE

The program specifies a two's complement Word Count and a Starting Address. When Start sets Busy, tape motion begins. Data is then transferred a word at a time to the controller from the Mini's Memory. The controller splits each word into two bytes and transfers them to the Drive one at a time. This operation continues until the Word Counter overflows or the Data Late condition occurs. If Write Protect was one, both Done and Illegal will set and no tape motion will result.

Possible Errors -

1. Unit not ready
2. Parity Error
3. Bad Tape
4. Data Late

6.6 WRITE END OF FILE

Start will set Busy, (unless Write Protect is on, which results in Done and Illegal setting). Tape motion will begin and a file mark will be written. Done sets when operation is complete.

6.7 ERASE

Start will set Busy (unless Write Protect is on, which results in Done and Illegal setting) and the next three inches of tape will be erased. Done sets when operation is complete.

7.0 DIAGNOSTICS AND TESTING

The Model 110 Coupler comes with a Diagnostic and Reliability Program both resident on a Cartridge shipped with the board. The programs may be found on the tape in the following format:

File 0	ABOOT	- Bootstrap Loader Program
File 1	DIRECTORY	- List of Programs on the Tape
File 2	MT110 DIAG	- Model 110 Coupler Diagnostic used for testing and or trouble shooting this specific board
File 3	UNIVERSAL MAG TAPE RELI	- Mag Tape Reliability used for determining Reliability of Data Written then Read
File 4		- All the previous files in a "DUMP" format for storage on a disk

7.1 TESTING

This board incorporates an On Board Self Test which is done automatically when the board is powered up. This test takes approximately 2 seconds for a successful completion. If an error exists on the board it will fail one of 5 tests. The test that has failed will be reported to the user by a blinking led found on the left hand side of the handle. For example - if Test 2 failed the led would blink 2 times, pause, and keep repeating the process. If no error is found, the led will be on for the duration of the Self Test. The Tests are:

7.1 TESTING (continued)

NUMBER OF BLINKS	TEST	DESCRIPTION
1	Stack Memory Test	- Checks stack and scratch pad memory for the Micro-processor
2	Done Test	- Checks Done logic for Data General hand shake
3	Increment Word Count Test	- Checks word counter for correct counting and that overflow conditions occur
4	PIO Test	- Checks PIO for good operation under interrupts
5	Checksum Test	- Makes sure that the programmed memory is correct

7.2 DIAGNOSTICS

If the board runs Self Test but does not run with the system, run MT110 Diagnostics. This test resides on File 2 of the diagnostic cartridge supplied with the board. Upon loading of the program (see Appendix E for Program Loading) the following message will be displayed:

- ....C.S.I..... MT110 Diag. (Kennedy 1/4") Rev.0X
- Mount a write enabled scratch tape, hit any key to cont

A Unit Request is made as follows:

- Drive Unit #
  Respond to the request by typing the number of the Unit you want to test. (0 or 1)

A Request for Device Code is made as follows:

- MTA Device Code
  Respond to the request by typing either 22 or 62.

A Request to Modify the Soft Switches is made as follows:

- Set switch register to desired value, type CR to continue.
  (Refer to diagnostic listing if you wish to modify.) (CR)

Run this Test until a Request is made to move Cartridge Switch to write lock position. It is done as follows:

- Move Cartridge Switch to the safe position
  Respond by not stopping the program, removing the cartridge, moving the switch and reinserting the tape.

A Request then will be made to return the cartridge to the write enabled position. This is done as follows:

- Move Cartridge Switch off the safe position.
  Reverse the previous instruction being careful not to stop the program.

The test will then proceed to write to end of tape and print a message as follows:

- End of Tape
- Cycle
- Pass 1

After one pass is completed no more testing is done on the write lock or EOT detection.

NOTE: To successfully run the tape diagnostic, the AOS Jumper (J4) must be in.

7.3 RELIABILITY

This program resides in File 3 of the Cartridge supplied with the board. The test reads and writes data in random record lengths, (Max 7 records), and random data patterns with all errors being reported. This process is continued through the entire data cartridge until EOT is sensed at which time the error history is displayed along with a pass count. Before the test can be run, the User must answer the questions in the Text. The questions are as follows:

```
- ....C.S.I..... Multi-Mag Tape Reli Rev. 0X

  STARTING ADDRESSES:
    500 - Reliability
    501 - Interchange Test (Write/Read)
    502 - Interchange Test (Read Only)
    503 - Command String Interpreter
    504 - Error Log Printout

Set Switch Register to desired value, Type CR to Continue
M
0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
0  0  0  0  0  0  0  0  0  0  0   0   0   0   0   0
   (Refer to Reliability Listing if you wish to modify)

Reliability Test
  Enter Device Code 22 or 62, followed by a CR  (22)
  Enter 0 to Test CRC (NRZI Only) 1 if not, followed by CR  (1)
  Mount Scratch Tape(s), enter CR to Continue (CR)
```

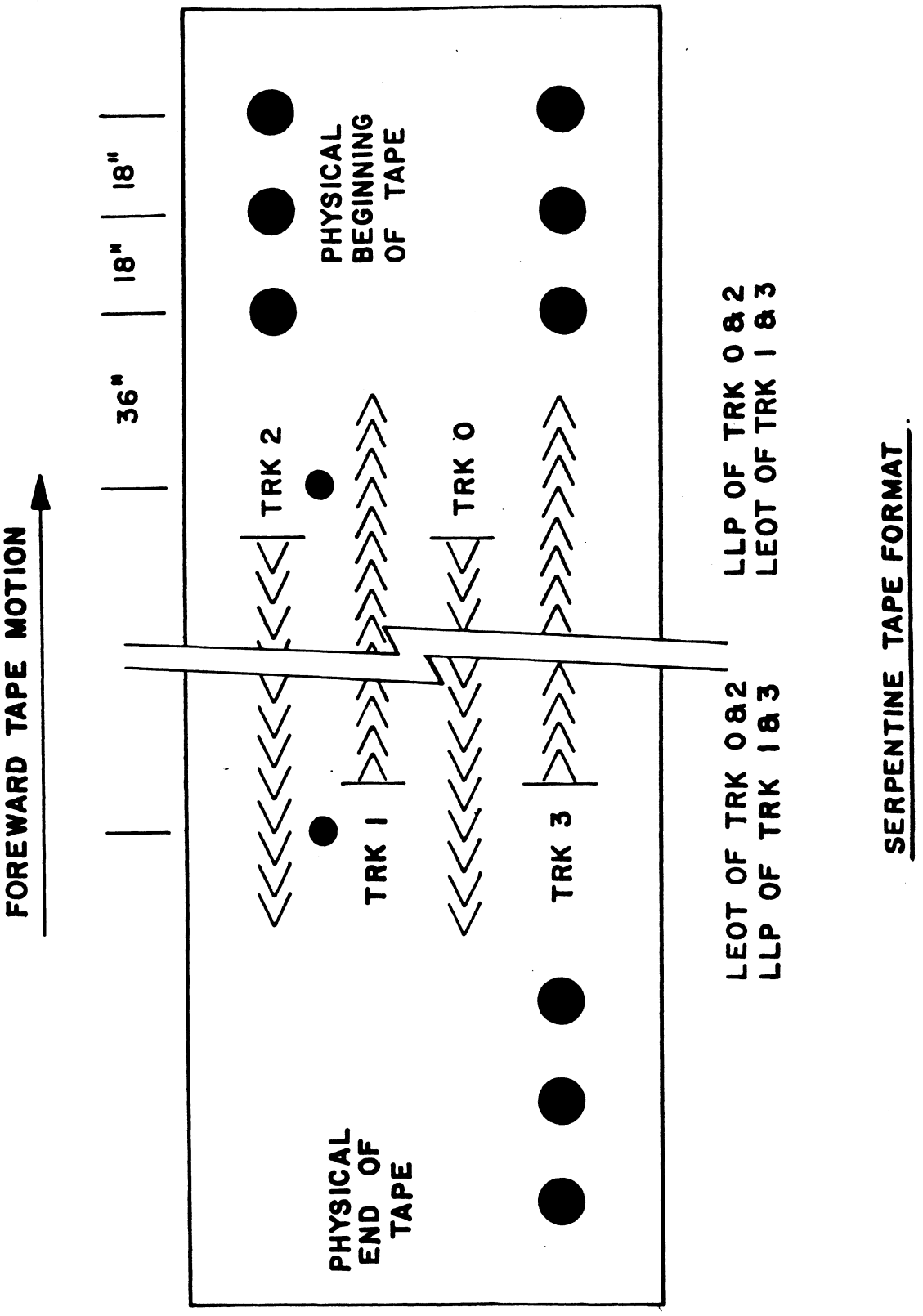
This test will print out any detected error whether it recovered or not. If it recovered the error, it will print how many passes it took to recover it. If it did not recover the error will print "Hard Error" after the error report.

APPENDIX A: PERFORMANCE CHART

RDOS UTILITY	DATA STORAGE	EXECUTION TIME IN MINUTES	RECORD SIZE	EFFECTIVE TRANSFER RATE	*EFFICIENCY
DUMP/LOAD	2.0 Megabytes	24.5 min.	512 bytes	1360 b/s	5.67%
FDUMP/FLOAD	9.5 Megabytes	9.75 min.	4096 bytes	16.24 Kb/s	67.67%
BURST DUMP/LOAD	10.5 Megabytes	9.5 min.	8192 bytes	18.42 Kb/s	76.8%

*Efficiency is a ratio of effective transfer rate divided by maximum transfer rate.

APPENDIX B: TAPE FORMAT



SERPENTINE TAPE FORMAT

DATA RECORD

GAP	PREAMBLE	SYNC	DATA FIELD	END MARK	CRCC	POSTAMBLE	GAP
1.2"	80 (1's)	00111		11111		80 (1's)	1.2"

FILE MARK

GAP	PREAMBLE	SYNC	END MARK	REV SYNC	POSTAMBLE	GAP
1.2"	80 (1's)	00111	11111	11100	80 (1's)	1.2"

APPENDIX D: CABLING PINOUTS

PIN #	J1	J2	DESCRIPTION	TO/FROM COUPLER
2	$\overline{\text{FCLR 0}}$	$\overline{\text{FCLR 1}}$	Formatter Clear	From
4	$\overline{\text{CREQ 0}}$	$\overline{\text{CREQ 1}}$	Command Request	From
6	$\overline{\text{DRDY 0}}$	$\overline{\text{DRDY 1}}$	Data Ready	From
8	$\overline{\text{HPAR 0}}$	$\overline{\text{HPAR 1}}$	Host Parity	From
10	$\overline{\text{STRB 0}}$	$\overline{\text{STRB 1}}$	Strobe	To
12	$\overline{\text{CBSY 0}}$	$\overline{\text{CBSY 1}}$	Command Busy	To
14	$\overline{\text{DBSY 0}}$	$\overline{\text{DBSY 1}}$	Data Busy	To
16	$\overline{\text{FPAR 0}}$	$\overline{\text{FPAR 1}}$	Formatter Parity	To
18	$\overline{\text{BUS0-0}}$	$\overline{\text{BUS1-0}}$	Data Bus 0	----
20	$\overline{\text{BUS0-1}}$	$\overline{\text{BUS1-1}}$	Data Bus 1	----
22	$\overline{\text{BUS0-2}}$	$\overline{\text{BUS1-2}}$	Data Bus 2	----
24	$\overline{\text{BUS0-3}}$	$\overline{\text{BUS1-3}}$	Data Bus 3	----
26	$\overline{\text{BUS0-4}}$	$\overline{\text{BUS1-4}}$	Data Bus 4	----
28	$\overline{\text{BUS0-5}}$	$\overline{\text{BUS1-5}}$	Data Bus 5	----
30	$\overline{\text{BUS0-6}}$	$\overline{\text{BUS1-6}}$	Data Bus 6	----
32	$\overline{\text{BUS0-7}}$	$\overline{\text{BUS1-7}}$	Data Bus 7	----
34	$\overline{\text{CMON 0}}$	$\overline{\text{CMON 1}}$	Cable Monitor	To

APPENDIX E: DIAGNOSTIC SUPPORT PACKAGE/GENERAL INFORMATION

BOOTSTRAP PROCEDURES
LOADING DSP FROM TAPE

1) Load DSP Tape Cartridge

2) Perform the following steps when the system has the program load option. (If system does not have program load option consult processor manual)

   A) Put 100022 or 100062 on console data switches 0-15.

   B) Program Load.

      1) Press program load switch if front panel has switches.
      2) DG virtual console, enter 100022L or 100062L (if 100062 first enter 100062 in 11A).
      3) Point 4 virtual console, set switches or CPU board, enter P22 or P62.

3) Enter tape file number, followed by a carriage return of desired test.

4) If program is not self starting perform the following steps:

   A) Front panel switches.

      1) Put starting address on console data switches (0-15).
      2) Press examine memory.
      3) Put switch settings on console data switches (0-15).
      4) Press continue.

   B) Virtual console.

      1) Front Panel Switches.

         A) Enter switch settings in 11A through keyboard.
         B) Enter starting address (XXXXR) through keyboard.
         C) To change switch settings, enter break, change 11A through keyboard, and enter PC address when break occurred. (XXXXR)
         D) To continue on error halt, enter PC address (XXXXR).

      2) Point 4.

         A) Set switches on CPU board.
         B) Enter starting address (JXXXXX).
         C) To continue on error halt, enter PC address (JXXXXX).

```
                ;
                ;
                ;
                ;*****************************************************************
                ;
                ;
                ;
                ; DESCRIPTION: MT110 1/4" MAGNETIC TAPE DIAGNOSTIC
                ;
                ;
                ; CUSTOM SYSTEMS INC,  1982
                ;*****************************************************************
                        .TITL   MT110D
   000001               .DUSR   X=1
   000001               .NOMAC  X
                ;1.      PROGRAM NAME: MT110D.SR
                ;2.      REVISION HISTORY:
                ;
                ;               REV.       DATE
22 000000               .REV    0.0     ;04/27/82
                ;
                ;3.      MACHINE REQUIREMENTS
                ;       3.1     NOVA OR ECLIPSE FAMILY CPU'S
                ;       3.2     MINIMUM OF 16K MEMORY
                ;       3.3     6450 KENNEDY 1/4" CARTRIDGE TAPE DRIVE
                ;       3.4     MT110 CONTROLLER BOARD
                ;       3.5     TELETYPE OR CRT AND CONTROLLER
                ;4.      TEST REQUIREMENTS
                ;       N/A
                ;5.      SUMMARY
                ;       THIS PROGRAM IS A HARDWARE DIAGNOSTIC FOR THE MODEL 110
                ;       CONTROLLER.  THE DEVICE CODE CAN BE SELECTED AS
                ;       22 OR 62.  ONLY ONE(1) READY, WRITE ENABLED DRIVE CAN BE
                ;       ON LINE AT A TIME.
                ;
                ;6.      RESTRICTIONS
                ;
                ;       ONLY ONE(1) DRIVE CAN BE ONLINE AT ANY TIME.  THE DEVICE
                ;       CODE MUST BE SELECTED AS 22 OR 62.  ALL RESPONSES TO PRO-
                ;       GRAM REQUESTS MUST BE ANSWERED PROPERLY TO CONTINUE THE
                ;       SEQUENTIAL TESTING OF THE TAPE DRIVE.
```

```
;7.      PROGRAM DESCRIPTION/THEORY OF OPERATION
;        7.1     INITIALIZATION
;                7.1.1   I/O MODULE INITIALIZED
;                7.1.2   TEST SELB LINE SET, IF LINE SET IR-
;                        RECOVERABLE ERROR.  PROGRAM HALTS AT
;                        BHALT.
;                                1.      SELECT UNIT NUMBER
;                                2.      DEVICE CODE CHANGE
;                                3.      SET SOFT SWITCH REGISTER
;        7.2     PRELIMINARY TESTS
;                7.2.1   TEST A1 - TEST SYSTEM SELD LINE.
;                7.2.2   TESTS A2 AND A3 - TEST CONTROLLER BUSY
;                        AND DONE STATUS.
;                7.2.3   TESTS A4 THRU A8 - TEST FOR UNIT SELECT
;                        BY LOADING AND TESTING THE MEMORY
;                        ADDRESS REGISTER.
;                7.2.4   TESTS A9 THRU A14 - TEST FOR UNIT READY
;                        AND THE SETTING AND RESETTING OF BUSY
;                        AND DONE BY THE START COMMAND.
;        7.3     FIRST TAPE MOTION
;                7.3.1   TESTS A15 AND A16 - TEST REWIND AND
;                        ERASE OPERATION AND STATUS.
;        7.4     FIRST DATA TRANSFER
;                7.4.4   TESTS A20 AND A21 - TEST FOR TOTAL DATA
;                        DATA WRITE WITH INTERRUPT.
;                7.4.5   TESTS A22 AND A23 - TEST WRITE EVEN AND
;                        ODD PARITY.
;        7.5     STATUS BIT TESTS
;                7.5.1   TEST A24 AND A25 - TEST FOR ILLEGAL
;                        COMMAND STATUS BIT SETTING.
;                7.5.2   TEST A26 - TEST FOR EOF STATUS BIT
;                        SETTING.
;                7.5.4   TESTS A28 THRU A 32 - TEST STATUS BITS
;                        AND MEMORY ADDRESS REGISTER DURING BACK
;                        AND FORWARD SPACING.
;        7.6     DATA TRANSFER TESTS
;                7.6.1   TESTS A33 AND A34 - TEST WRITE AND READ
;                        IN ODD AND EVEN PARITY.
;                7.6.2   TESTS A35 AND A36 - TEST WRITE AND READ
;                        WITH DIFFERENT WORD COUNTS.
;                7.6.4   TESTS A39 THRU A41 - TEST EOF WRITE AND
;                        READ.
;                7.6.6   TESTS A50 THRU A53 - TEST FOR SPACING
;                        ERRORS BY GENERATING NOISE WITH I/O
;                        COMMANDS.
;                7.6.7   TEST A54 - TEST LONGITUDINAL PARITY
;                        ERRORS BY WRITTING EVEN PARITY, ZERO
;                        DATA PATTERNS.
;        7.7     WRITE LOCK TEST
;                THIS TEST DETERMINES IF THE CARTRIDGE SET TO SAFE
;                WILL DISABLE THE WRITE.  THIS TEST IS ONLY
;                PERFORMED DURING THE FIRST PASS AND CAN BE DE-
;                LETED BY SETTING SOFT SWITCH REGISTER BIT 15.
;        7.8     END OF TAPE TEST
;                THIS TEST WRITES 4K BLOCKS FROM BOT TO EOT.  DUR-
;                ING THE TAPE WRITE ALL ERROR STATUS CONDITIONS
;                ARE MONITORED.  WHEN THE EOT SENSOR IS DETECTED
;                THE WRITE OPERATION IS TERMINATED AND THE TAPE IS
;                COMMANDED TO REWIND.  IF THE EOT SENSOR IS NOT DE-
;                TECTED THE WRITE WILL CONTINUE UNTIL THE TAPE
```

```
;               COMES OFF THE SUPPLY REEL.  THIS TEST CAN BE DE-
;               LETED BY SETTING SOFT SWITCH REGISTER BIT 14.
;8.     SOFT SWITCH REGISTER SETTINGS
       S?WPD 8
;      8.3     SWITCH OPTIONS
;
;               DIFFERENT SWITCH BITS AND THEIR INTERPRETATION
;                AT LOCATION "SWREG" ARE AS FOLLOWS:
;
;               BIT     OCTAL   BINARY  INTERPRETATION
;                       VALUE   VALUE
;               14(E)   00002   0       ENABLE WRITE TO EOT TEST
;                               1       INHIBIT WRITE TO EOT TEST
;
;               15(F)   00001   0       ENABLE WRITE LOCK TEST
;                               1       INHIBIT WRITE LOCK TEST
;
;               NOTE:   SWITCH BITS 14 AND 15 CAN ONLY BE
;                       ENABLED DURING THE FIRST PASS OF THE
;                       DIAGNOSTIC.  IF THE TESTS ARE TO BE PER-
;                       FERMED AFTER THE FIRST PASS, THEY CAN BE
;                       DIRECTLY ENTERED.
;9.     OPERATING PROCEDURES
;      9.1     PROGRAM LOAD
;              LOAD THE PROGRAM BY USING THE BINARY LOADER.
;      9.2     STARTING ADDRESSES
;              201     DIRECT ENTRY TO OCTAL DEBUGGER(ODT)
;              500     START DIAGNOSTIC
;              501     DIRECT ENTRY TO WRITE LOCK TEST
;              502     DIRECT ENTRY TO WRITE TO EOT TEST
;      9.3     PROGRAM OPERATION
;              9.3.1   THE FOLLOWING HEADER IS PRINTED.
;      "C.S.I.  MT110 DIAG. (KENNEDY 1/4") REV.  XX"
;      "MOUNT A WRITE ENABLED SCRATCH TAPE, HIT ANY KEY TO CONTINUE"
;              9.3.2   A UNIT REQUEST IS MADE AS FOLLOWS:
;      "DRIVE UNIT #:"
;                      RESPOND TO THE REQUEST BY TYPING A
;                      UNIT NUMBER 0 OR 1 FOLLOWED BY A
;                      CARRIAGE RETURN.
;              9.3.5   A REQUEST FOR A DEVICE CODE IS MADE
;                      AS FOLLOWS:
;      "MTA DEVICE CODE:"
;                      RESPOND TO THE REQUEST BY TYPING EITHER
;                      22 OR 62.  ANY OTHER INPUT WILL CAUSE AN
;                      ERROR.
;              9.3.6   A REQUEST TO MODIFY THE SOFT SWITCH
;                      REGISTER IS MADE AS FOLLOWS:
;      "SET SWITCH REGISTER TO DESIRED VALUE, TYPE CR TO CONTINUE."
;                      THE PROGRAM WILL BE LOCKED IN THE MODIF-
;                      ICATION MODE.  SET ALL BITS TO THEIR COR-
;                      RECT STATE BY HITTING THE APPROPRIATE
;                      KEY.  TO TERMINATE THE SWITCH MODIFICATION
;                      MODE TYPE A CARRIAGE RETURN.
;              9.3.7   IF A SYSTEM REAL TIME CLOCK IS NOT PRE-
;                      SENT, THE FOLLOWING REQUEST WILL BE
;                      PRINTED.
;      "TTO BAUD RATE = ? "
;                      RESPOND TO THE REQUEST BY TYPING THE
;                      CORRECT CONSOLE DEVICE BAUD RATE FOR THE
;                      I/O TIMING CALIBRATION.  IF THE RESPONSE
```

```
;                       IS 110, THE FOLLOWING REQUEST MESSAGE
;                       WILL BE PRINTED.
;       "(ONLY DASHER AT 110 BAUD = 10 BITS/CHAR"
;       "OTHERWISE = 11 BITS/CHAR.)"
;       "# BITS/CHAR = ? "
;                       RESPOND TO THE REQUEST BY TYPING 10 OR
;                       11.
;               9.3.8   THE DIAGNOSTIC TESTING WILL TEST THE
;                       CONTROLLER AND SELECTED DRIVE.  ALL DE-
;                       TECTED FAULTS WILL BE PRINTED AS SPEC-
;                       IFIED BY SECTION 10 AND PROGRAM CON-
;                       TROL IS DETERMINEDBY THE SETTING OF THE
;                       SOFT SWITCH REGISTER.  EACH SUCCESSIVE
;                       TEST ASSUMES THAT ALL PREVIOUS TESTS
;                       HAVE PASSED CORRECTLY.  BYPASSING A
;                       FAILING TEST MAY LEAD TO CONFUSING
;                       RESULTS ON SUCCEDING TESTS.
;
;               9.3.9   WRITE LOCK TEST
;                       THE WRITE LOCK TEST IS ENTERED DURING
;                       THE FIRST PASS OF THE DIAGNOSTIC OR BY
;                       DIRECT ENTRY.  THE SELECTED UNIT IS
;                       TESTED BY ATTEMPTING TO WRITE WHEN THE
;                       CARTRIDGE IS SET TO SAFE.  THE FOLLOW-
;                       ING REQUEST IS PRINTED AT THE START OF
;                       THE TEST.
;       "MOVE CARTRIDGE SWITCH TO THE SAFE POSITION.  DONT STOP THE PROGRAM."
;                       RESPOND TO THE REQUEST AS FOLLOWS:
;                               MOVE CARTRIDGE SWITCH TO SAFE
;                       WHEN THE DRIVE COMES READY, AN ATTEMPT
;                       IS MADE TO WRITE A RECORD.  IF THE WRITE
;                       IS INHIBITED, THE FOLLOWING REQUEST IS
;                       PRINTED.
;       "MOVE CARTRIDGE SWITCH OFF THE SAFE POSITION."
;                       MOVE SWITCH OFF THE SAFE POSITION
;               9.3.10  WRITE TO EOT SENSOR
;                       THE TEST IS PERFORMED DURING THE FIRST
;                       PASS OF THE DIAGNOSTIC OR BY DIRECT
;                       ENTRY.  THE TEST WRITES 4K DATA BLOCKS
;                       FROM BOT TO THE EOT SENSOR.  ALL STATUS
;                       BITS ARE EXAMINED DURING EACH WRITE AND
;                       IF ANY FAULTS ARE DETECTED AN APPROPRI-
;                       ATE STATUS ERROR MESSAGE IS PRINTED.  IF
;                       THE EOT SENSOR IS NOT DETECTED THE WRITE
;                       WILL CONTINUE UNTIL THE TAPE COMES OFF
;                       THE SUPPLY REEL INDICATING THE ERROR.  IF
;                       THE EOT SENSOR IS DETECTED, THE FOLLOW-
;                       ING MESSAGE IS PRINTED.
;       "END OF TAPE"
;       "CYCLE"
;                       AFTER THE MESSAGE IS PRINTED, THE DRIVE
;                       IS COMMANDED TO REWIND.  WHEN THE REWIND
;                       OPERATION IS COMPLETE THE FOLLOWING MES-
;                       SAGE IS PRINTED.
;       "PASS 1"
;10.    PROGRAM ERROR DESCRIPTION
;       10.1    PRELIMINARY TEST ERRORS
;               THE FOLLOWING IS A LIST OF PRELIMINARY CON-
;               TROLLER AND DRIVE ERROR MESSAGES.
;               10.1.1  BUSY AND DONE ERRORS
```

```
;       "SELD LINE NOT RESET BY IORST, PC = XXXXX"
;       "BUSY FLIP-FLOP NOT RESET ERROR, PC = XXXXX"
;       "BUSY FLIP-FLOP NOT RESET BY IORST, PC = XXXXX"
;       "BUSY FLIP-FLOP NOT SET ERROR, PC = XXXXX"
;       "DONE FLIP-FLOP NOT RESET ERROR, PC = XXXXX"
;       "DONE FLIP-FLOP NOT SET ERROR, PC = XXXXX"
;              10.1.2  CONTROLLER DATA TRANSFER ERRORS
;       "SEND CLOCK BIT ON TOO LONG ERROR, PC = XXXXX"
;       "FIRST CHARACTER TIME OUT ERROR, PC = XXXXX"
;       "DATA TRANSFER TIME OUT ERROR, PC = XXXXX"
;       "NO INTERRUPT ERROR, PC = XXXXX"
;       "ILLEGAL INTERRUPT WITH MASK BIT SET, MASK = XX,/
;       PC = XXXXX"
;       "MTU SELECT ERROR, DIB COMMAND = XXXXXX, PC = XXXXX"
;       "MA REGISTER NOT RESET BY IORST"
;       "GOOD WORD = XXXXXX, BAD WORD = XXXXXX, PC = XXXXX"
;       "MA REGISTER SETTING ERROR"
;       "GOOD WORD = XXXXXX, BAD WORD = XXXXXX, PC = XXXXX"
;       "INTA DEVICE CODE ERROR"
;       "DEVICE CODE = XX, UNIT DEVICE CODE = XX, PC = XXXXX"
;       10.2    SYSTEM ERRORS
;              THE FOLLOWING ERRORS OCCURE DURING COMBINED CON-
;              TROLLER AND DRIVE OPERATIONS.
;              10.2.1  DATA TRANSFER AND MA REGISTER ERRORS
;       "MA REGISTER COUNTING ERROR"
;       "GOOD VALUE = XXXXXX, BAD VALUE = XXXXXX, PC = XXXXX"
;       "DATA COMPARE ERROR"
;       "GOOD WORD = XXXXXX, BAD WORD = XXXXXX, /
;       MEMORY ADDRESS = XXXXXX, PC = XXXXX"
;              10.2.2  STATUS ERRORS
;       "EXPECTED STATUS = XXXXXX, ACTUAL STATUS = XXXXXX, /
;       PC = XXXXX"
;       10.3    STATUS WORD
;              BIT     DESCRIPTION
;              0       ANY ERROR, SET BY BITS 1,3,5,6,7,8,10,14
;              1       DATA LATE
;              2       REWINDING
;              3       ILLEGAL COMMAND
;              4       HIGH DENSITY
;              5       PARITY ERROR
;              6       EOT MARK SENSED
;              7       EOF MARK SENSED
;              8       BOT MARK SENSED
;              9       9 TRACK TAPE
;              10      BAD TAPE
;              11      SEND CLOCK
;              12      FIRST CHARCTER
;              13      WRITE LOCKOUT
;              14      CRC ERROR
;              15      UNIT READY
        O?DTD   11
;12.    SPECIAL NOTES
;       12.1    MEDIA SELECTION
;              IT IS IMPORTANT TO SELECT A KNOWN GOOD TAPE WHEN
;              PERFORMING THE DIAGNOSTIC.  ANY ERRORS CAUSED BY
;              THE MEDIA WILL BE CONSIDERED A CONTROLLER AND/OR
;              DRIVE FAULT.
;       12.2    SCOPE LOOPS
;              WHEN A SCOPE LOOP IS BEING IMPLEMENTED TO LOCATE
;              A FAILING MODULE AND FORWARD TAPE MOTION IS
```

```
;               USED, THE TAPE WILL COME OFF THE SUPPLY REEL IF
;               THE LOOP IS ALLOWED TO CONTINUE.  WHEN THE TAPE
;               APPROACHES THE EOT SENSOR, ENTER THE ODT PROGRAM
;               BY TYPEING A CONTROL "O" CHARACTER, MANUALLY RE-
;               WIND THE DRIVE AND TYPE A "P" CHARACTER TO CON-
;               TINUE.
;13.     RUN TIME
;       THE PROGRAM RUN TIME IS DEPENDENT ON THE
;       LENGTH OF THE TAPE.
;       IT IS RECOMMENDED THAT A
;       300 FOOT CARTRIDGE BE USED TO FACILITATE A FAST
;       WRITE TO THE EOT SENSOR TEST.
```

```
   000000          .NOMAC  0
   000000          .LOC 0
00 000000          0
01 000010          INTR
02 000200          DTO?SB
03 002002          JMP     @.-1
04 000000          0
05 000000          0

          ; IRRECOVERABLE ERROR HALT, SELB LINE ALWAYS SET

06 063077 BHALT:   HALT                    ; HALT FOR HARD SELB LINE SET
07 000200          JMP     DTO?SB          ; START PROGRAM AGAIN ON CONTINUE
10 010000 INTR:    ISZ     0
11 002000          JMP     @0
12 002226          JMP     @IODT?          ;GO TO ODT

   000045          .LOC    45              ;DO NOT INSERT,LOC 45
                                           ;CONTAINS EGGS POINTER
45 006757 NEST:    EGGS

   000050          .LOC    50

50 000010          .BLK    10              ; RESERVED FOR SYSTEM DEBUG

                   ;CONSTANTS
60 000503 FIRST:   DIAG
61 007016 CEND:    LAST
62 075402 CNTA:    DIB 3,2
63 007016 OBUFF:   LAST
64 007216 IBUFF:   LAST+200
65 000031 .25MS:   25.
66 000062 .50MS:   50.
67 000310 .200MS:  200.
70 000620 .400MS:  400.
71 010421 C10421:  10421
72 021042 C21042:  21042
73 042104 C42104:  42104
74 004210 C421T:   4210
75 160077 C160077:         160077
76 003265 PCDCM:   CDCM
77 177700 C177700:         177700
00 110000 C110000:         110000
01 125252 C125252:         125252
02 052525 C52525:  52525
03 000144 DL100:   100.
04 001064 CRCNT:   500.+100

05 000001 C1:      1
06 000002 C2:      2
07 000003 C3:      3
10 000004 C4:      4
11 000005 C5:      5
12 000006 C6:      6
13 000007 C7:      7
14 000010 C10:     10
15 000011 C11:     11
16 000012 C12:     12
17 000013 C13:     13
```

```
MT110
20 000014 C14:    14
21 000015 C15:    15
22 000016 C16:    16
23 000017 C17:    17
24 000020 C20:    20
25 000024 C20.:   20.
26 000021 C21:    21
27 000022 C22:    22
30 000030 C30:    30
31 000040 C40:    40
32 000060 C60:    60
33 000062 C62:    62
34 000070 C70:    70
35 000077 C77:    77
36 000100 C100:   100
   000103 C100.=DL100
37 000101 C101:   101
40 000102 C102:   102
41 000103 C103:   103
42 000104 C104:   104
43 000144 C144:   144
44 000150 C150:   150
45 000200 C200:   200
46 000201 C201:   201
47 001000 C1000:  1000
50 002000 C2000:  2000
51 020000 C20K:   20000
52 100401 C1004:  100401
53 004000 C4K:    4000
54 007700 C7700:  7700
55 037477 C37477: 37477
56 040000 C40000: 40000
   000150 C2MS=C2000

57 005670 D3000:  3000.
60 000764 D500:   500.

61 177775 M3:     -3
62 177774 M4:     -4
63 177770 M8:     -8.
64 177700 M100:   -100
65 177160 M400:   -400.
66 000116 ASCN:   116
67 000131 ASCY:   131
70 000000 IIDX0:  0               ;%
71 000000 IIDX1:  0               ;%
72 000000 IIDX2:  0               ;%
73 000000 IIDX3:  0               ;%
74 000000 IIDX4:  0               ;%

        ; POINTERS AND CONSTANTS

   000200         .LOC 200

   000000         .DUSR   COM?P=0

200 002202 DTO?SB: JMP @BGNADR            ;PROGRAM START
201 002226         JMP @IODT?             ;DIRECT ENTER TO ODT
202 000500 BGNADR: BEGIN                  ; PROGRAM STARTING ADDRESS
203 000000 PAS?S:  0                      ; PASS COUNTER
```

```
MT110
004 000000 ODO?K:  0                       ; ODT BREAKPOINT LOCATION

          ; PAGE ZERO POINTERS

005 006764 ISWR?EG: SWREG                   ; SWITCH REGISTER POINTER
006 005750 IINP?:  INP?J                    ; SWITCH PACK POINTER
007 005226 IMES?:  MES?S                    ; MESSAGE PRINT POINTER
010 005275 ICRL?F: CRL?F                    ; CR/LF PRINT ROUTINE POINTER
011 005424 ITYP?E: TYP?E                    ; CHARACTER PRINT ROUTINE POINTER
012 005330 IPDE?C: PDE?C                    ; DECIMAL PRINT ROUTINE POINTER
013 005320 IPDC?S: PDC?S                    ; DECIMAL AND 1 CHAR. PRINT
014 005312 IPOC?T: POC?T                    ; OCTAL PRINT ROUTINE POINTER
015 005306 IZOC?T: ZOC?T                    ; ZERO SURPRESSED OCTAL PRINT
016 005570 ITI?O:  TIN?O                    ; OCTAL INPUT ROUTINE POINTER
017 005574 ITI?D:  TIN?D                    ; DECIMAL INPUT ROUTINE POINTER
020 005421 ITPS?P: TPS?P                    ; TYPE SPACE ROUTINE POINTER
021 000000 IOM?OD: 0                        ; I/O MODULE POINTER
022 005670 ITI?A:  TIN?A                    ; INPUT CHARATER ROUTINE POINTER
023 006420 ICAL?B: CAL?B                    ; CALIBRATE TIME BASE POINTER
024 006624 IDEL?Y: DEL?Y                    ; TIME DELAY POINTER
025 006642 ITIM?R: TIM?R                    ; TIMER TEST POINTER
026 006212 IODT?:  ODT?J                    ; OCTAL DEBUGGER POINTER
027 006721 IMLT?:  MUL?T                    ; MULTIPLY ROUTINE POINTER
030 006741 IDVI?:  DIV?I                    ; DIVIDE ROUTINE POINTER
031 005533 IPC7?:  PC7?                     ; TAB COUNTER POINTER

                   ;VARIABLES

032 000000 X70:    0
033 000010 C1X:    10                       ;UNIT # +10
034 000000 CX:     0                        ;UNIT #
035 000050 C5X:    50                       ;UNIT # +50
036 000000 CC5X:   0                        ;UNIT # +50
037 000070 C7X:    70                       ; UNIT # + 70
040 000000 CNTR:   0                        ;COUNTER
041 000000 WDCNT:  0                        ;TEMP STORAGE
042 000000 INST:   0                        ;  "     "
043 000022 DEVICE: .MTA                     ;DEVICE CODE.  22 OR 62
044 000000 RECTR:  0
045 000000 WCTR:   0

                   ;PATTERN TABLE, ODD PARITY WRITES

046 000246 POINT:  .
047 000000         0
050 177400         177400
051 000377         377
052 125252         125252
053 052525         52525
054 000253 LST:    .-1

                   ;PATTERN TABLE, EVEN PARITY WRITES

055 000255 PONTR:  .
056 125252         125252
057 052525         052525
060 000401         000401
061 020040         020040
062 000261 LST1:   .-1
```

```
                    ;INDIRECT ADDRESSES
 63 003165 ICKST:   CSTAT
 64 002431 INTIL:   INITIAL
 65 003207 ISTAT:   XSTAT
 66 003144 IERAS:   XERAS
 67 003072 IWRT:    XWRT
 70 002763 IGEN:    XGEN
 71 002722 IWEOF:   XWEOF
 72 002615 IBSP:    XBSP
 73 003067 IWNS:    XWNS
 74 004557 ABTBF:   BTBF
 75 004617 SETP1:   ENTER-3
 76 004622 SETP:    ENTER
 77 004640 LOOP.:   CYCLE
 00 004722 ER:      ERR
 01 003015 IRD:     XRD
 02 002577 ICLR:    XCLR
 03 002547 ICHK:    XCHK
 04 002743 ILD:     XLD
 05 002620 ISPC:    XSPC
 06 003002 ISEL:    XSEL
 07 003274 PINWL:   INWLT
 10 003233 PETTA:   ETTA
 11 003266 IWLT:    WLT
 12 003224 IETT:    ETT
 13 005202 ITEST:   TSTIN
 14 004714 ISPE1:   SPER1
 15 004717 ISPE2:   SPER2
                    ;DEFINITIONS
    000022 .MTA=22
    006265 STATUS=JSR @ISTAT
    006266 ERASE=JSR @IERAS
    006267 WRITE=JSR @IWRT
    006270 GEN=JSR @IGEN
    006271 WEOF=JSR @IWEOF
    006272 BSPACE=JSR @IBSP
    006273 WRTNS=JSR @IWNS
    006224 WAIT=JSR @IDEL?Y
    006305 SPACE=JSR @ISPC
    006300 EHALT=JSR @ER
    006314 HALT1=JSR @ISPE1
    006315 HALT2=JSR @ISPE2
    006301 READ=JSR @IRD
    006302 CLEAR=JSR @ICLR
    006303 CHECK=JSR @ICHK
    006304 LOAD=JSR @ILD
    006306 SELECT=JSR @ISEL
316 002656 .RWNS
    006316 RWNS=JSR @.-1
317 002661 .RW
    006317 REWIND=JSR @.-1
320 002614 .XBSP
    006320 BSPNS=JSR @.-1
    006207 MESSAGE=JSR @IMES?S
    006215 TYPZO=JSR @IZOC?T
    006216 OCTIN=JSR @ITI?O
    006222 ASCIN=JSR @ITI?A
    006214 TYPEO=JSR @IPOC?T
    006223 CALIB=JSR @ICAL?IB
    006212 TYPZD=JSR @IPDE?C
```

```
MT110
   006211 TYPEA=JSR @ITYP?E
   006212 TYPED=JSR @IPDE?C

   000121 ASCR=C15
   000123 CONO=C17
```

```
MT110
   000500          .LOC    500

00 000403 BEGIN:  JMP     DIAG            ; START DIAGNOSTIC HERE
01 002311         JMP @IWLT               ;WRITE LOCK TEST
02 002312         JMP @IETT               ;END TAPE TEST

03 062677 DIAG:   IORST                   ; CLEAR ALL I/O
04 030045         LDA     2,NEST          ; GET THE POINTER
05 021000         LDA     0,0,2           ; GET I/O MODULE FLAG
06 101005         MOV     0,0,SNR         ; TEST FOR FLAG SET
07 000411         JMP     DIAG1           ; IF NOT SET, SET POINTER
10 021001         LDA     0,1,2           ; GET EGGS + 1
11 024135         LDA     1,C77           ; GET MASK BITS
12 124000         COM     1,1             ; SET TO MASK INSTRUCTION
13 123525         ANDZL   1,0,SNR         ; ISOLATE I/O MOD
14 000404         JMP     DIAG1           ; IF NOT SET, CLEAR POINTER
15 025004         LDA     1,4,2           ; GET TOP OF MEMORY
16 106400         SUB     0,1             ; SUBTRACT MOD FROM TOP
17 121400         INC     1,0             ; MEMORY TOP + 1
20 040221 DIAG1:  STA     0,IOM?OD        ; SET I/O MOD POINTER

          ; START DIAGNOSTIC HERE

21 102520         SUBZL   0,0             ; SET REGISTER TO 1
22 042231         STA     0,@IPC7?        ; SET TAB TO ONE SPACE
23 102400         SUB     0,0             ; CLEAR REGISTER
24 042205         STA     0,@ISWR?EG      ; INITIALIZE SWREG

                  ;CHECK SELB BUS LINE

25 062677 A0:     IORST                   ; CLEAR I/O SYSTEM
26 063500         SKPBZ   0               ; TEST FOR SELB LINE SET
27 000006         JMP     BHALT           ; IF SET, IRRECOVERABLE ERROR
30 006264         JSR     @INTIL          ; GO INITIALIZE SYSTEM

                  ;CHECK SELD BUS LINE

31 006276 A1:     JSR @SETP               ;DO SKPDZ, DEVICE 0
32 062677         IORST                   ; CLEAR I/O SYSTEM
33 006224         WAIT                    ; WAIT FOR CONSOLE INPUT
34 000067         .200MS
35 152520         SUBZL   2,2             ; SET = 1 FOR ERROR PRINT
36 063700         SKPDZ   0               ; TEST FOR SELD LINE SET
37 006314         HALT1                   ; IF SET, GO PROCESS ERROR
40 006277         JSR @LOOP.

                  ;CHECK MTA BUSY FLOP OFF

41 006276 A2:     JSR     @SETP           ;DO BUSY TEST ON
42 030106         LDA     2,C2            ; SET = 2 FOR ERROR PRINT
43 063522         SKPBZ   .MTA            ;MTA
44 006314         HALT1                   ;CANNOT RESET MT BUSY
45 006277         JSR     @LOOP.

                  ;CHECK MTA DONE FLOP OFF

46 006276 A3:     JSR     @SETP           ;DONE TEST ON MTA
47 030111         LDA     2,C5            ; SET = 5 FOR ERROR PRINT
50 063722         SKPDZ   .MTA
51 006314         HALT1                   ;CANNOT RESET MT DONE
```

```
MT110
52 006277          JSR     @LOOP.

                   ;CHECK MT SELECT,DOB,DIB,ADDR REG.  IN-OUT

53 006276 A4:      JSR     @SETP           ;LOAD CA WITH 125252
54 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
55 020101          LDA     0,C125252       ;READ BACK AND CHECK
56 062022          DOB     0,.MTA
57 065422          DIB     1,.MTA
60 101120          MOVZL   0,0
61 101220          MOVZR   0,0             ;AC0=VALUE SENT TO ADDR REG
62 106414          SUB#    0,1,SZR         ;AC1=VALUE READ BACK
63 006314          HALT1                   ;CHECK THE MT SELECT, LOAD ADDRESS
64 006277          JSR     @LOOP.          ;ENABLE ADDR. A&B, AND ADDR. REG.

                   ;CHECK CA REGISTER IN&OUT

65 006276 A5:      JSR     @SETP           ;LOAD CA WITH 052525
66 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
67 020102          LDA     0,C52525        ;READ BACK AND CHECK
70 062022          DOB     0,.MTA
71 065422          DIB     1,.MTA
72 106414          SUB#    0,1,SZR         ;AC0=VALUE SENT TO ADDR REG
73 006314          HALT1                   ;AC1=VALUE READ BACK
74 006277          JSR     @LOOP.          ;CHECK ADDR REG DATA PATHS

                   ;CHECK FOR ILLEGAL MT SELECT

75 024062 A6:      LDA     1,CMTA          ;LOAD CA WITH 052525
76 044404          STA     1,A6.2          ;READ BACK USING EVERY
77 006275 A6.1:    JSR     @SETP1          ;DEVICE CODE EXCEPT MTA
00 020102          LDA     0,C52525        ;DATA SHOULD NOT BE
01 062022          DOB     0,.MTA          ;FOUND!
02 075402 A6.2:    DIB     3,2             ;THIS IS CHANGED BY PROG
03 030120          LDA     2,C14           ; SET = 14 FOR ERROR PRINT
04 162415          SUB#    3,0,SNR
05 006314          HALT1                   ;AC0=DATA SENT TO ADDR REG
                                           ;AC1=DATA READ BACK
                                           ;PROBLEM IS IN MT SELECT
06 006277          JSR     @LOOP.          ;THE DEVICE SELECT GATE.
07 024773          LDA     1,A6.2
10 125400 A6.3:    INC     1,1
11 020135          LDA     0,C77
12 123405          AND     1,0,SNR
13 000406          JMP     A7              ;DONE ALL GO ON
14 030243          LDA     2,DEVICE
15 142415          SUB#    2,0,SNR
16 000772          JMP     A6.3            ;MTA, INC AGAIN
17 044763          STA     1,A6.2
20 000757          JMP     A6.1

                   ;CHECK CA REGISTER, ALL DATA PATTERNS

21 102400 A7:      SUB     0,0             ;LOAD CA, READ BACK
22 006275          JSR     @SETP1          ;AND CHECK.  USE
23 062022          DOB     0,.MTA          ;ALL FROM 0-077777
24 065422          DIB     1,.MTA
25 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
26 106414          SUB#    0,1,SZR         ;AC0=DATA SENT TO ADDR REG
27 006314          HALT1                   ;AC1=DATA READ BACK
```

```
MT110
030 006277          JSR     @LOOP.          ;ADDR REG IS PATTERN SENSITIVE
031 101400          INC     0,0
032 101113          MOVL#   0,0,SNC
033 000767          JMP     A7+1
034 000401          JMP     A8

                    ;CHECK IO RESET OF CA REGISTER

035 006276 A8:      JSR     @SETP           ;LOAD CA WITH ALL 1'S
036 102000          ADC     0,0             ;DO IORST,  READ
037 062022          DOB     0,.MTA          ;CA AND CHECK FOR 0'S
040 062677          IORST
041 006224          WAIT                    ; WAIT FOR CONSOLE INPUT
042 000067          .200MS
043 030121          LDA     2,C15           ; SET = 15 FOR ERROR PRINT
044 065422          DIB     1,.MTA          ;AC0=DATA SENT TO ADDR REG
045 125004          MOV     1,1,SZR         ;AC1=DATA READ BACK
046 006314          HALT1                   ;CHECK IORST, RESET, AND
047 006277          JSR     @LOOP.          ;ADDRESS REGISTER RESET GATES

                    ;CHECK CA LOAD

050 006276 A8A:     JSR     @SETP           ;LOAD CA WITH 010421
051 102400          SUB     0,0             ;LOAD CA WITH ALL ZEROS
052 024071          LDA     1,C10421        ;READ AND CHECK
053 066022          DOB     1,.MTA
054 062022          DOB     0,.MTA
055 065422          DIB     1,.MTA
056 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
057 125004          MOV     1,1,SZR         ;CA LOADING ERROR
060 006314          HALT1                   ;AC1=(CA)
061 006277          JSR     @LOOP.          ;(CA) SHOULD BE ZERO

062 006276 A8B:     JSR     @SETP           ;LOAD CA WITH 21042
063 102400          SUB     0,0             ;LOAD CA WITH ALL ZEROS
064 024072          LDA     1,C21042        ;READ AND CHECK
065 066022          DOB     1,.MTA
066 062022          DOB     0,.MTA
067 065422          DIB     1,.MTA
070 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
071 125004          MOV     1,1,SZR         ;CA LOADING ERROR
072 006314          HALT1                   ;AC1=(CA)
073 006277          JSR     @LOOP.          ;(CA) SHOULD BE ZERO

074 006276 A8C:     JSR @SETP               ;LOAD CA WITH 42104
075 102400          SUB     0,0             ;LOAD CA WITH ALL ZEROS
076 024073          LDA     1,C42104
077 066022          DOB     1,.MTA
700 062022          DOB     0,.MTA
701 065422          DIB     1,.MTA
702 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
703 125004          MOV     1,1,SZR         ;CA LOADING ERROR
704 006314          HALT1                   ;AC1=(CA)
705 006277          JSR     @LOOP.          ;(CA) SHOULD BE ZERO

706 006276 A8D:     JSR     @SETP           ;LOAD CA WITH 4210
707 102400          SUB     0,0             ;LOAD CA WITH ALL ZEROS
710 024074          LDA     1,C421T         ;READ AND CHECK
711 066022          DOB     1,.MTA
712 062022          DOB     0,.MTA
```

```
MT110
13 065422          DIB     1,.MTA
14 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
15 125004          MOV     1,1,SZR         ;CA LOADING ERROR
16 006314          HALT1                   ;AC1=(CA)
17 006277          JSR     @LOOP.          ;(CA) SHOULD BE ZERO

                   ;CHECK CA LOAD

20 006276 A8E:     JSR     @SETP
21 024071          LDA     1,C10421
22 034072          LDA     3,C21042
23 030073          LDA     2,C42104
24 020074          LDA     0,C421T
25 066022          DOB     1,.MTA
26 076022          DOB     3,.MTA
27 072022          DOB     2,.MTA
30 062022          DOB     0,.MTA
31 065422          DIB     1,.MTA
32 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
33 106414          SUB#    0,1,SZR         ;CA LOADING ERROR
34 006314          HALT1                   ;AC1=(CA)
35 006277          JSR     @LOOP.          ;(CA) SHOULD BE 004210

                   ;CHECK FOR "TUR"

'36 006276 A9:     JSR     @SETP           ;FIRST ATTEMPT TO
'37 006306         SELECT                  ;READ STATUS
'40 126520         SUBZL   1,1             ; SET READY BIT
'41 060422         DIA     0,.MTA          ;TAPE UNIT READY (TUR)
'42 107415         AND#    0,1,SNR         ; IS BIT 15.
'43 006300         EHALT                   ;CHECK DRIVE FOR THE
'44 006277         JSR     @LOOP.          ;READY STATE.

                   ;CHECK MT START AND BUSY FLOP

'45 006276 A9A:    JSR     @SETP           ;SEND START PULSE
'46 006306         SELECT
'47 030111         LDA     2,C5            ; SET = 5 FOR ERROR PRINT
'50 060122         NIOS    .MTA            ;SKIP ON BUSY SET
'51 063622         SKPDN   .MTA            ;MT START DID NOT CLEAR DONE
'52 000403         JMP     A9B             ; IF CLEAR, TEST BUSY
'53 006314         HALT1
'54 000404         JMP     A9C             ; TEST LOOP ON ERROR
'55 030110 A9B:    LDA     2,C4            ; SET = 4 FOR ERROR PRINT
'56 063422         SKPBN   .MTA            ;MT START DID NOT SET BUSY
'57 006314         HALT1                   ;CHECK MT START,BUSY,DONE FLOPS
'60 006277 A9C:    JSR     @LOOP.          ;AND SET LINE

                   ;CHECK IORST OF BUSY FLOP (CLEAR COM)

'61 006276 A10:    JSR     @SETP           ;SEND START PULSE
'62 006306         SELECT
'63 060122         NIOS    .MTA
'64 062677         IORST
'65 006224         WAIT                    ; WAIT FOR CONSOLE INPUT
'66 000067         .200MS
'67 030107         LDA     2,C3            ; SET = 3 FOR ERROR PRINT
'70 063522         SKPBZ   .MTA            ;IORST DIDN'T RESET BUSY
'71 006314         HALT1                   ;CHECK CLEAR COM AND
'72 006277         JSR     @LOOP.          ;MT BUSY RESET LINE
```

```
                ;CHECK CLEAR PULSE

'73 006276 A11:  JSR     @SETP          ;SEND START PULSE
'74 006306       SELECT
'75 060122       NIOS    .MTA
'76 060222       NIOC    .MTA
'77 030111       LDA     2,C5           ; SET = 5 FOR ERROR PRINT
100 063622       SKPDN   .MTA           ;MT CLR DIDN'T RESET DONE
101 000403       JMP     A11A           ; IF RESET, TEST BUSY
102 006314       HALT1                  ; IF SET, PROCESS ERROR
103 000404       JMP     A11B           ; GO LOOP ON ERROR
104 030106 A11A: LDA     2,C2           ; SET = 2 FOR ERROR PRINT
105 063522       SKPBZ   .MTA
106 006314       HALT1                  ;MT CLR DIDN'T RESET BUSY
107 006277 A11B: JSR     @LOOP.         ;CHECK MT CLR BUSY,DONE FLOPS

                ;CHECK DOA-MTLC-COMMAND DECODER

110 006275 A12:  JSR     @SETP1         ;SELECT REWIND
111 006306       SELECT                 ;SEND START PULSE
112 030106       LDA     2,C2           ; SET = 2 FOR ERROR PRINT
113 020233       LDA     0,C1X          ;MAKE SURE BUSY DOESN'T SET
114 061122       DOAS    0,.MTA         ;CHECK E29 GATE DRIVING THE
115 063522       SKPBZ   .MTA           ;MT BUSY SET LINE.  REWIND IS ON
116 006314       HALT1                  ;AND MT START IS SETTING BUSY.
117 006277       JSR     @LOOP.         ;CHECK COMMAND DECODER, E2.

                ;CHECK DIA AND STATUS REGISTER AT BOT

120 006275 A13:  JSR     @SETP1         ;REWIND TAPE
121 006306       SELECT                 ;READ AND CHECK STATUS
122 006317       REWIND
123 006265       STATUS                 ;AC1=EXPECTED STATUS, AC0=BAD.
124 100201       100201
125 006300       EHALT                  ;CHECK RD STATUS, REWIND THRU TO
126 000401       JMP     .+1            ;TAPE, AND BOT LINE FORM TAPE.
127 006277       JSR     @LOOP.         ;ALSO BOT STATUS REG.  GATE

                ;CHECK UNIT SELECT LOGIC, ERASE COMMAND DECODE

130 020134 A14:  LDA     0,C70          ;SELECT EACH UNIT
131 040232       STA     0,X70          ;IN SEQUENCE 0-7
132 006276 A14.1: JSR    @SETP          ;CHECK FOR TUR IF
133 020232       LDA     0,X70          ;UNIT IS SELECTED.
134 061022       DOA     0,.MTA         ;OTHERWISE CHECK FOR
135 006263       JSR     @ICKST         ;NO TUR.
136 006300       EHALT                  ;AC1=EXPECTED STATUS, AC0=BAD
                                        ;AC2 13-15=UNIT JUST TESTED
                                        ;IF UNIT TESTED=UNIT SELECTED
                                        ;PROCEED ON THE STATUS INFO.
137 006277       JSR     @LOOP.         ;IF NOT CHECK FOR ERRONEOUS TUR.

140 010232       ISZ     X70
141 020232       LDA     0,X70
142 024136       LDA     1,C100
143 122414       SUB#    1,0,SZR
144 000766       JMP     A14.1

                ;CHECK FOR "TUR" FOLLOWING ERASE
```

```
006275 A14A:    JSR     @SETP1          ;REWIND TO START TEST
006317          REWIND                  ;AT BOT
006266          ERASE                   ;ERASE & WAIT 1 SEC
006265          STATUS
000001          1                       ;AC1=EXPECTED STATUS
006300          EHALT                   ;AC0=ACTUAL STATUS
000401          JMP     .+1             ;FOLLOWING ERASE
006277          JSR     @LOOP.

                ;CHECK FOR "DONE" FOLLOWING ERASE

006275 A14B:    JSR     @SETP1          ;REWIND TO START TEST
006317          REWIND                  ;AT BOT
006266          ERASE                   ;ERASE & TIME 1 SEC
030106          LDA     2,C2            ; SET = 6 FOR ERROR PRINT
063422          SKPBN   .MTA
000403          JMP     A14C            ; IF NOT BUSY, TEST DONE
006314          HALT1                   ; IF SET, PROCESS ERROR
000404          JMP     A14D            ; TEST FOR LOOP ON ERROR
030112 A14C:    LDA     2,C6            ; SET = 6 FOR ERROR PRINT
063622          SKPDN   .MTA            ;BUSY=1 OR DONE=0
006314          HALT1                   ;FOLLOWING ERASE
006277 A14D:    JSR     @LOOP.

                ;CHECK REWIND-ERASE-REWIND SEQUENCE

006275 A15:     JSR     @SETP1          ;REWIND WAIT 5 SEC.
006317          REWIND                  ;ERASE WAIT 1 SEC.
006266          ERASE                   ;CHECK STATUS (TUR)
006265          STATUS                  ;REWIND-CHK STATUS
000001          1
006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
000406          JMP     A15E            ;IF (BOT), ERASE DIDN'T MOVE TAPE.
                                        ;CHECK ERASE COMMAND THRU TO TAPE
006317          REWIND
006265          STATUS
100201          100201                  ;AC1=EXPECTED STATUS, AC0=BAD.
006300          EHALT                   ;REWIND FOLLOWING ERASE DIDN'T
000401          JMP     .+1             ;WORK.  THE CLUE IS IN THE BAD
006277 A15E:    JSR     @LOOP.          ;STATUS.

                ;CHECK REWINDING STATUS BIT

006275 A16:     JSR     @SETP1          ;DO ERASE-RWNS
006266          ERASE                   ;CHECK STATUS
006316          RWNS                    ;DURING REWIND
024151          LDA     1,C20K
060422          DIA     0,.MTA
123405          AND     1,0,SNR
006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
006317          REWIND                  ;CHECK REWINDING STATUS BIT GATE
006277          JSR     @LOOP.


                ;CHECK SIMPLE 4 CHAR WRITE

006276 A20:     JSR     @SETP           ;WRITE 2WD
006270          GEN                     ;CHECK CA, DONE FLAG
000377 A20.1:   377
```

```
MT110
 22 006267          WRITE
 23 000102          102
 24 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
 25 063722          SKPDZ   .MTA            ;DONE FLAG SHOULD BE ON.
 26 000403          JMP     .+3             ;THIS IS THE FIRST COMPLETE WRITE
 27 006314          HALT1                   ;AND INVOLVES ALL THE WRITE
 30 000416          JMP     A20E            ;LOGIC.  TO COMPLETE A WRITE
                                            ;ERF-LOAD DELAY-DLY OVER-STOP-
 31 024063          LDA     1,OBUFF         ;TUR-MT DONE, CLEAR MT BUSY.
 32 030106          LDA     2,C2            ;CHECK WC OVER, INCREMENT WC.
 33 147000          ADD     2,1
 34 061422          DIB     0,.MTA
 35 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
 36 106415          SUB#    0,1,SNR
 37 000403          JMP     .+3             ;ADDR REG. WRONG!  AC1=GOOD
 40 006314          HALT1                   ;AC0=BAD.  TWO OUTPUT XFERS
 41 000405          JMP     A20E            ;SHOULD HAVE OCCURRED
 42 006265          STATUS
 43 000001          1
 44 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
 45 000401          JMP     .+1             ;DATA PATTERN IS 1-0-1-0
                                            ;FOR EACH TRACK.  LATERAL
 46 006277 A20E:    JSR     @LOOP.          ;PARITY IS ALWAYS 1.

                    ;CHECK INTERRUPT AND DISABLE

 47 006275 A21:     JSR     @SETP1          ;WRITE 2 WDS
 50 006267          WRITE
 51 000102          102
 52 030116          LDA     2,C12           ; SET = 12 FOR ERROR PRINT
 53 060177          INTEN                   ;ENABLE INTERRUPT
 54 000401          JMP     .+1             ;ALLOW 1 INST BEFORE INT.
 55 006314          HALT1                   ;NO INTERRUPT.  CHECK MT INT REQ.
 56 006277          JSR     @LOOP.          ;INTR GATE, INTP IN GATE.

                    ;CHECK PROPER RESPONSE TO INTA

 57 006275 A21A:    JSR     @SETP1          ;WRITE 2 WORDS
 60 006267          WRITE
 61 000102          102
 62 030124          LDA     2,C20           ; SET = 20 FOR ERROR PRINT
 63 061477          INTA    0               ;READ DEVICE ADDR
 64 024243          LDA     1,DEVICE
 65 122414          SUB#    1,0,SZR         ;AC1=CORRECT INTA RESPONSE
 66 006314          HALT1                   ;AC0=BAD
 67 006277          JSR     @LOOP.

                    ;INSURE THAT MSKO WILL DISABLE INTERRUPT

170 006275 A21B:    JSR     @SETP1          ;WRITE 2 WORDS
171 006267          WRITE
172 000102          102
173 024131          LDA     1,C40
174 066077          MSKO    1               ;DISABLE MTA INTERRUPTS
175 030117          LDA     2,C13           ; SET = 13 FOR ERROR PRINT
176 060177          INTEN                   ;ALLOW INTERRUPTS
177 000401          JMP     .+1
200 000402          JMP     .+2             ;NOT PERFORMED IF INTERRUPT
201 006314          HALT1                   ;ILLEGAL INTERRUPT.  CHECK
202 060277          INTDS                   ;MT INT DISABLE FLOP,
```

```
MT110
033 006277          JSR     @LOOP.          ;AMD INPUTS
                    ;WRITE 16 CHAR RECORD.  ODD PARITY
                    ;PATTERN IN A22.2

034 102000 A22:     ADC     0,0
035 040405          STA     0,A22.2
036 020246          LDA     0,POINT
037 040171          STA     0,IIDX1
040 006275 A22.1:   JSR     @SETP1          ;WRITE
                                            ;CHECK DONE, CA
041 006270          GEN                     ;STATUS.
042 177777 A22.2:   177777
043 006267          WRITE
044 000110          110
045 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
046 063722          SKPDZ   .MTA            ;NO DONE FLAG!  RECORDS TO NOW
047 000403          JMP     .+3             ;HAVE BEEN 4 CHAR.  THIS ONE IS
050 006314          HALT1                   ;16 CHAR.  THIS IS THE ONLY
051 000416          JMP     A22E            ;DIFFERENCE.  READ AFTER WRITE
                                            ;DETECTS ERF,AND ERF-LOAD DELAY-
052 024063          LDA     1,OBUFF         ;DLY OVER-STOP-TUR-MT DONE.
053 030114          LDA     2,C10           ;CHECK WC OVER, INCREMENT WC
054 147000          ADD     2,1
055 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
056 051422          DIB     0,.MTA
057 106415          SUB#    0,1,SNR         ;CA WRONG.  AC1=CORRECT
060 000403          JMP     .+3             ;AC0=BAD.  CHECK DATA
061 006314          HALT1                   ;CHAN CONTROL AND INCREMENT
062 000405          JMP     A22E            ;WC.  10 OCTAL WORDS WRITTEN

063 006265          STATUS
064 000001          1
065 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
066 000401          JMP     .+1             ;THE CLUE IS IN THE BAD STATUS
067 006277 A22E:    JSR     @LOOP.

070 020171          LDA     0,IIDX1
071 024254          LDA     1,LST
072 106405          SUB     0,1,SNR
073 000405          JMP     A23             ;GO TO NEXT TEST
074 010171          ISZ     IIDX1           ;%
075 022171          LDA     0,@IIDX1
076 040744          STA     0,A22.2         ;SET NEW DATA PATTERN
077 000741          JMP     A22.1
                    ;WRITE 16 CHAR RECORD.  EVEN PARITY
                    ;PATTERN IS IN A23.2

150 102000 A23:     ADC     0,0
151 040405          STA     0,A23.2
152 020255          LDA     0,PONTR
153 040171          STA     0,IIDX1
154 006275 A23.1:   JSR     @SETP1          ;WRITE
                                            ;CHECK DONE, CA
155 006270          GEN                     ;STATUS
156 177777 A23.2:   177777
157 006267          WRITE
160 100110          100110
161 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
162 063722          SKPDZ   .MTA            ;NO DONE FLAG!  THIS HAS WORKED
163 000403          JMP     .+3             ;MANY TIMES BEFORE IN ODD PAR.
```

```
MT110
64 006314          HALT1                ;CHECK READ AFTER WRITE ERF
65 000416          JMP     A23E         ;DETECTION.
66 024063          LDA     1,OBUFF
67 030114          LDA     2,C10
70 147000          ADD     2,1
71 030126          LDA     2,C21        ; SET = 21 FOR ERROR PRINT
72 061422          DIB     0,.MTA
73 106415          SUB#    0,1,SNR      ;ADDR. REG. WRONG.  AC1=CORRECT
74 000403          JMP     .+3          ;AC0=BAD.  CHECK DATA CHAN REQ.
75 006314          HALT1                ;AND INCREMENT WC.
76 000405          JMP     A23E
77 006265          STATUS
00 000001          1
01 006300          EHALT                ;AC1=CORRECT STATUS, AC0=BAD
02 000401          JMP     .+1          ;PARITY ERRORS ARE LIKELY DUE
                                        ;TO THE PARITY CHECKING LOGIC
03 006277 A23E:    JSR     @LOOP.       ;(EVEN PARITY)

04 020171          LDA     0,IIDX1
05 024262          LDA     1,LST1
06 106405          SUB     0,1,SNR
07 000405          JMP     A24
10 010171          ISZ     IIDX1        ;%
11 022171          LDA     0,@IIDX1
12 040744          STA     0,A23.2
13 000741          JMP     A23.1

                   ;TEST ILLEGAL STATUS

14 034234 A24:     LDA     3,CX         ;GIVE START TO UNSELECTED
15 030105          LDA     2,C1         ; TAPE
16 175400          INC     3,3          ;AC2=UNIT
17 173400          AND     3,2
20 006276          JSR     @SETP
21 071022          DOA     2,.MTA       ;START READ
22 060122          NIOS    .MTA
23 063522          SKPBZ   .MTA
24 000777          JMP     .-1
25 060422          DIA     0,.MTA       ;READ STATUS
26 024100          LDA     1,C110000    ;AC1=THE 2 BITS EXPECTED
27 123400          AND     1,0          ;AC0=THE 2 BITS FOUND
30 122414          SUB#    1,0,SZR      ;LOOKING FOR ERROR AND ILLEGAL
31 006300          EHALT                ;STATUS.  CHECK ILLEGAL AND SET
                                        ;GATES (E31), ALSO STATUS BIT
32 006277          JSR     @LOOP.       ;GATE (E52).
                   ;TEST ILLEGAL STATUS

33 006275 A25:     JSR     @SETP1       ;REWIND.  GIVE
34 006317          REWIND               ;SPACE BACK AT BOT
35 006272          BSPACE
36 000001          1
37 006265          STATUS
40 110201          110201               ;AC1=EXPECTED STATUS, AC0=BAD.
41 006300          EHALT                ;CHECK SPACE BKD&BOT GATE ON
42 000401          JMP     .+1          ;ILLEGAL (E31).  CHECK SPACE BKD
43 006277          JSR     @LOOP.       ;COMMAND DECODE

                   ;CHECK "DONE" FOLLOWING WRITE EOF

44 006275 A26A:    JSR     @SETP1
```

```
;T110
45 006271          WEOF                    ;WRITE EOF
46 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
47 063622          SKPDN   .MTA
50 006214          HALT1                   ;NO DONE FLAG
51 006277          JSR     @LOOP.          ;ON WRITE EOF

                   ;CHECK WRITE EOF STATUS

52 006276 A26:     JSR     @SETP           ; WEOF
53 006271          WEOF                    ;AND CHECK STATUS
54 006265          STATUS
55 100401          100401
56 006300          EHALT                   ;AC1=EXPECTED STATUS.  AC0=BAD
57 000401          JMP     .+1             ;CHECKING FOR EOF,ERROR.   CHECK EOF
60 006277          JSR     @LOOP.          ;FLOP AND C-D INPUTS.


61 006275 A28:     JSR     @SETP1          ;REWIND-WRITE 1 REC.
62 006317          REWIND                  ;BACKSPACE AND CHECK
63 006267          WRITE                   ;DONE AND CORRECT
64 000102          102                     ;STATUS
65 006272          BSPACE
66 000001          1
67 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
70 063722          SKPDZ   .MTA            ;NO DONE FLAG FOLLOWING
71 000403          JMP     .+3             ;BACKSPACE.   CHECK SPACE BKD
72 006314          HALT1                   ;COMMAND DECODE AND DRIVER
73 000405          JMP     A28E            ;TO TAPE (E20)
74 006265          STATUS
75 000001          1
76 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
77 000401          JMP     .+1
00 006277 A28E:    JSR     @LOOP.

                   ;CHECK BACKSPACE

01 006275 A30:     JSR     @SETP1          ;REWIND-CHECK FOR BOT
02 006317          REWIND                  ;WRITE 2 REC.   CHECK
03 006265          STATUS                  ;STATUS, BACKSPACE 2
04 100201          100201                  ;CHECK STATUS AT BOT
05 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD.
06 000416          JMP     A30E            ;LOOKING FOR EOT FOLLOWING
                                           ;REWIND.   THIS SHOULD WORK!
07 006267          WRITE
10 000202          202
11 006272          BSPACE
12 000002          2
13 006265          STATUS
14 000001          1
15 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD.
16 000406          JMP     A30E
17 061422          DIB     0,.MTA
20 024140          LDA     1,C102
21 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
22 122414          SUB#    1,0,SZR         ;ADDRESS REG. ERROR.
23 006314          HALT1                   ;2 REC SHOULD ICR IT
24 006277 A30E:    JSR     @LOOP.          ;TWICE DURING BACKSPC
                   ;CHECK BACKSPACE

25 006275 A31:     JSR     @SETP1          ;REWIND-WRITE 3 REC.
```

```
MT110
026 006317          REWIND                  ;BACKSPACE 4, CHECK
027 006267          WRITE                   ;BOT AND CA FOR
030 000302          302                     ;3 COUNTS
031 006272          BSPACE
032 000004          4
033 006265          STATUS
034 100201          100201
035 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
                                            ;BOT SHOULD STOP OPERATION
                                            ;IN 3 BACKSPACES
036 006277          JSR     @LOOP.

                    ;CHECK SPACE FORWARD

037 006275 A32:     JSR     @SETP1          ;REWIND-WRITE 3 REC.
040 006317          REWIND                  ;REWIND-CHECK STATUS
041 006267          WRITE                   ;SPACE FORWARD-CHECK
042 000302          302                     ;FOR NO BOT, & PROPER
043 006317          REWIND                  ;ADDR. REG.
044 006265          STATUS
045 100201          100201
046 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
047 000414          JMP     A32E            ;THIS HAS WORKED MANY TIMES BEFORE
050 006305          SPACE
051 000001          1
052 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD.
053 000001          1                       ;THIS IS THE FIRST SAPCE FORWARD.
054 006300          EHALT                   ;CHECK COMMAND DECODER-SPACE FWD-
055 000406          JMP     A32E            ;COUNT WC (E43)
056 024137          LDA     1,C101
057 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
060 051422          DIB     0,.MTA          ;ADDR REG WRONG.  AC1=CORRECT
061 106414          SUB#    0,1,SZR         ;AC0=ACTUAL.  ONE INCR FROM 100
062 006314          HALT1                   ;TO 101 SHOULD HAVE OCCURRED.
                                            ;CHECK SPACE FWD NOT-COUNT WC-
063 006277 A32E:    JSR     @LOOP.          ;INCREMENT WC.

                    ;CHECK READ INSTRUCTION.  READ 16 CHAR
                    ;RECORDS.  PATTERN IN A33.2 (ODD PARITY)

064 102000 A33:     ADC     0,0             ;REWIND-WRITE 1 REC.
065 040406          STA     0,A33.2         ;CHECK STATUS-REWIND-
066 020246          LDA     0,POINT         ;CHECK STATUS-READ-
067 040171          STA     0,IIDX1         ;CHECK DONE,STATUS,
070 006275 A33.1:   JSR     @SETP1          ;ADDR. REG., & DATA
071 006317          REWIND                  ;BUFFER
072 006270          GEN
073 017777 A33.2:   17777
074 006267          WRITE
075 000110          110
076 006265          STATUS
077 000001          1
100 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
101 000436          JMP     A33E            ;THIS MUCH HAS WORKED BEFORE,
102 006317          REWIND
103 006265          STATUS
104 100201          100201                  ;AC1=EXPECTED STATUS, AC0=BAD
105 006300          EHALT                   ;LOOKING FOR BOT BEFORE TRYING
106 000431          JMP     A33E            ;FIRST READ.  THIS HAS WORKED BEFORE
107 006302          CLEAR
```

```
.T110
000301          READ
000110          110
030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
063722          SKPDZ   .MTA            ;NO DONE FLAG FOLLOWING READ.
000403          JMP     .+3             ;THIS IS THE FIRST READ INST.
006314          HALT1                   ;CHECK READ-DCHI-INCREMENT WC
000421          JMP     A33E
006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
000001          1                       ;90% OF THE READ LOGIC HAS BEEN
006300          EHALT                   ;CHECKED DURING WRITE OPERATIONS.
000415          JMP     A33E            ;PROBLEMS HERE ARE LIKELY IN THE
                                        ;PROCESSOR INTERFACE AREA.
020064          LDA     0,IBUFF
024114          LDA     1,C10
107000          ADD     0,1
061422          DIB     0,.MTA
030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
106415          SUB#    0,1,SNR         ;ADDRESS REGISTER IS WRONG.
000403          JMP     .+3             ;AC1=CORRECT, AC0=ACTUAL.
006314          HALT1                   ;10 WORD REC READ.  CHECK
000404          JMP     A33E            ;MT DCH SEL-DCHM0-INCREMENT WC
006303          CHECK                   ;INPUT BUFFER CONTAINS WRONG
000010          10                      ;DATA.  INPUT BUFFER SHOULD =
006315          HALT2                   ;OUTPUT BUFFER FOR THE FIRST 8
                                        ;WORDS.  AC0=GOOD WORD, AC1=BAD
006277 A33E:    JSR     @LOOP.          ;AC2=ADDRESS OF INPUT BUFFER

020171          LDA     0,IIDX1
024254          LDA     1,LST
106405          SUB     0,1,SNR
000405          JMP     A34
010171          ISZ     IIDX1           ;%
022171          LDA     0,@IIDX1
040725          STA     0,A33.2
000721          JMP     A33.1

                ;CHECK READ INSTRUCTION.  READ 16 CHAR
                ;RECORDS.  PATTERN IN A34.2 (EVEN PARITY)

102000 A34:     ADC     0,0             ;REWIND-WRITE 1 REC.
040406          STA     0,A34.2         ;CHECK STATUS-REWIND
020255          LDA     0,PONTR         ;CHECK STATUS-READ
040171          STA     0,IIDX1         ;CHECK DONE,STATUS,
006275 A34.1:   JSR     @SETP1          ;ADDRESS REG.,DATA IN
006317          REWIND                  ;BUFFER
006270          GEN
177777 A34.2:   177777
006267          WRITE
100110          100110
006265          STATUS
000001          1
006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
000436          JMP     A34E            ;THIS MUCH HAS WORKED BEFORE!
006317          REWIND
006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
100201          100201                  ;LOOKING FOR BOT STATUS BEFORE
006300          EHALT                   ;TRYING FIRST EVEN PARITY READ.
000431          JMP     A34E            ;THIS SHOULD WORK!
006302          CLEAR
006301          READ
```

```
MT110
075 100110          100110
076 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
077 063722          SKPDZ   .MTA
000 000403          JMP     .+3
001 006314          HALT1                   ;NO DONE FLAG.  CHECK ERF-
002 000421          JMP     A34E            ;DCHI-INCREMENT WC
003 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
004 000001          1                       ;CHECK PARITY CIRCUITS.  THIS
005 006300          EHALT                   ;IS THE ONLY DIFFERENCE FROM
006 000415          JMP     A34E            ;THE LAST TEST.
007 020064          LDA     0,IBUFF
010 024114          LDA     1,C10
011 107000          ADD     0,1
012 061422          DIB     0,.MTA
013 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
014 106415          SUB#    0,1,SNR         ;ADDRESS REG. IS WRONG.  AC1=
015 000403          JMP     .+3             ;CORRECT VALUE, AC0=ACTUAL.
016 006314          HALT1                   ;10 WORD REC. READ.  CHECK MT
017 000404          JMP     A34E            ;DCH SEL-DCHM0-INCREMENT WC
020 006303          CHECK                   ;INPUT BUFFER CONTAINS WRONG
021 000010          10                      ;DATA.  INPUT BUFFER SHOULD
022 006315          HALT2                   ;=OUTPUT BUFFER FOR FIRST 8 WORDS
                                            ;AC0=GOOD WORD, AC1=BAD WORD
023 006277 A34E:    JSR     @LOOP.          ;AC2=ADDRESS OF THE INPUT BUFFER

024 020171          LDA     0,IIDX1
025 024262          LDA     1,LST1
026 106405          SUB     0,1,SNR
027 000405          JMP     A35
030 010171          ISZ     IIDX1           ;%
031 022171          LDA     0,@IIDX1
032 040725          STA     0,A34.2
033 000721          JMP     A34.1

                    ;READ 16 CHAR REC.  WITH WC=63

034 006275 A35:     JSR     @SETP1          ;REWIND- WRITE 1 REC.
035 006317          REWIND                  ;CHECK STATUS-REWIND
036 006270          GEN                     ;READ-CHECK DONE,
037 125252          125252                  ;STATUS,ADDR REG,
040 006267          WRITE                   ;AND DATA
041 000110          110
042 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
043 000001          1                       ;DON'T TYR TO READ IT IF IT
044 006300          EHALT                   ;CANNOT BE WRITTEN PROPERLY.
045 000431          JMP     A35E            ;THIS HAS WORKED BEFORE,
046 006317          REWIND
047 006301          READ
050 000177          177
051 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
052 063722          SKPDZ   .MTA
053 000403          JMP     .+3
054 006314          HALT1                   ;NO DONE FLAG.  CHECK INCREMENT
055 000421          JMP     A35E            ;WC-ERF-STOP-TUR
056 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
057 000001          1                       ;THIS IS THE FIRST TIME ON A
060 006300          EHALT                   ;READ THAT THE WC DIDN'T MATCH
061 000415          JMP     A35E            ;THE RECORD LENGTH.  CHECK
                                            ;ERF GATE ON STOP.
062 020064          LDA     0,IBUFF
```

```
.MT110
 53 024114          LDA     1,C10
 54 107000          ADD     0,1
 55 061422          DIB     0,.MTA
 56 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
 57 106415          SUB#    0,1,SNR         ;ADDRESS REGISTER WRONG
 60 000402          JMP     .+3             ;AC1=CORRECT ADDRESS, AC0=BAD.
 61 006314          HALT1                   ;CHECK INCREMENT WC
 62 000404          JMP     A35E
 63 006303          CHECK
 64 000010          10
 65 006315          HALT2                   ;DATA ERROR.
                                            ;AC0=GOOD WORD,  AC1=BAD WORD
 66 006277 A35E:    JSR     @LOOP.          ;AC2=ADDRESS OF THE INPUT BUFFER
                    ;READ 63 CHAR REC.  WITH WC=10

 67 006275 A36:     JSR     @SETP1          ;REWIND-WRITE 1 REC.
 70 006317          REWIND                  ;CHECK STATUS-REWIND
 71 006270          GEN                     ;READ-CHECK DONE,
 72 125252          125252                  ;STATUS,ADDR.  REG.
 73 006267          WRITE                   ;AND DATA
 74 000177          177
 75 006265          STATUS
 76 000001          1
 77 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
 00 000431          JMP     A36E            ;THIS MUCH HAS WORKED BEFORE
 01 006317          REWIND
 02 006301          READ
 03 000110          110
 04 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
 05 063722          SKPDZ   .MTA
 06 000403          JMP     .+3
 07 006314          HALT1                   ;NO DONE FLAG.  CHECK INCREMENT
 10 000421          JMP     A36E            ;WC-WC OVERFLOW-STOP
 11 006265          STATUS
 12 000001          1                       ;AC1=EXPECTED STATUS, AC0=BAD
 13 006300          EHALT                   ;THIS IS THE FIRST TIME A READ IS
 14 000415          JMP     A36E            ;STOPPED BY WC OVERFLOW ALONE.
                                            ;LOOK FOR WC OVERFLOW AND CHECK
                                            ;THE WC OVERFLOW GATE ON STOP
 15 020064          LDA     0,IBUFF
 16 024114          LDA     1,C10
 17 107000          ADD     0,1
 20 061422          DIB     0,.MTA
 21 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
 22 106415          SUB#    0,1,SNR         ;ADDRESS REG.  WRONG.  AC1=CORRECT
 23 000403          JMP     .+3             ;ADDR.  AC0=BAD.
 24 006314          HALT1                   ;CHECK DCHI LOGIC AND INCREMENT WC
 25 000404          JMP     A36E            ;DATA ERROR.  INPUT BUFFER SHOULD
 26 006303          CHECK                   ;=OUTPUT BUFFER FOR THE FIRST 8
 27 000010          10                      ;WORDS.
 30 006315          HALT2                   ;AC0=GOOD WORD,  AC1=BAD WORD
 31 006277 A36E:    JSR     @LOOP.          ;AC2=ADDRESS OF THE INPUT BUFFER
 32 006275 A39:     JSR     @SETP1          ;REWIND-WRITE A 16
 33 006317          REWIND                  ;CHAR REC. , WRITE EOF
 34 006267          WRITE                   ;CHECK STATUS
 35 000110          110                     ;REWIND-READ 2 REC.
 36 006271          WEOF                    ;CHECK STATUS,CA
 37 006265          STATUS
 40 100401          100401
 41 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
```

```
MT110
52 000420          JMP     A39E            ;THIS HAS WORKED BEFORE.
53 006217          REWIND
54 006201          READ
55 000210          210
56 024152          LDA     3,C1004
57 054402          STA     3,.+2
60 006265          STATUS
61 100401          100401
62 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
63 000407          JMP     A39E            ;LOOKING FOR READ EOF.
64 024064          LDA     1,IBUFF
65 125400          INC     1,1
66 061422          DIB     0,.MTA
67 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
70 106414          SUB#    0,1,SZR         ;ADDRESS REG WRONG.  AC1=CORRECT
71 006314          HALT1                   ;AC0=BAD.   EOF SHOULD INCR
72 006277 A39E:    JSR     @LOOP.          ;WC AND ADDR. BY 1

                   ;CHECK READ EOF ON SPACE BACK

73 006275 A40:     JSR     @SETP1          ;REWIND-WRITE 2 REC
74 006217          REWIND                  ;WRITE EOF THEN 3 REC
75 006267          WRITE                   ;BACKSPACE 10
76 000210          210                     ;CHECK STATUS AND
77 006271          WEOF                    ;ADDR REG.
00 006265          STATUS
01 100401          100401
02 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
03 000416          JMP     A40E            ;THIS HAS WORKED BEFORE.
04 006267          WRITE
05 000310          310
06 006272          BSPACE
07 000010          10
10 006265          STATUS
11 100401          100401
12 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
13 000406          JMP     A40E            ;LOOKING FOR EOF.   CHECK
                                           ;EOF GATE ON STOP
14 024142          LDA     1,C104
15 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
16 061422          DIB     0,.MTA          ;ADDRESS REG.  IS WRONG.
17 106414          SUB#    0,1,SZR         ;AC1=CORRECT VALUE, AC0=BAD
20 006314          HALT1                   ;CHECK "COUNT WC" AND INCREMENT
                                           ;SHOULD GET 4 COUNT WC
21 006277 A40E:    JSR     @LOOP.          ;BEFORE STOP
                   ;CHECK READ EOF ON SPACE FORWARD

22 006275 A41:     JSR     @SETP1          ;REWIND-WRITE 2 REC
23 006217          REWIND                  ;WEOF-WRITE 3 REC
24 006267          WRITE                   ;REWIND-SPACE 5
25 000210          210                     ;CHECK STATUS,ADDR
26 006271          WEOF                    ;REG.
27 006265          STATUS
30 100401          100401                  ;AC1=EXPECTED STATUS, AC0=BAD
31 006300          EHALT                   ;THIS HAS WORKED BEFORE.
32 000417          JMP     A41E
33 006267          WRITE
34 000310          310
35 006217          REWIND
36 006305          SPACE
```

```
.T110
37 000005          5
40 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
41 100401          100401                  ;LOOKING FOR EOF STATUS.  THIS IS
42 006300          EHALT                   ;THE SAME AS THE LAST EXCEPT THAT
43 000406          JMP     A41E            ;SPACING IS FORWARD
44 024141          LDA     1,C103
45 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
46 061422          DIB     0,.MTA          ;ADDRESS ERROR.  AC1=CORRECT
47 106414          SUB#    0,1,SZR         ;AC0=BAD.  TWO INCREMENT WC
50 006314          HALT1                   ;SHOULD HAVE OCCURRED
51 006277 A41E:    JSR     @LOOP.          ;EOF GIVES STOP.
```

```
                ; SPACING CHECK
                ; DURING SPACING MANY I/O COMMANDS ARE
                ; ISSUED TO GENERATE NOISE

052 006275 A50:    JSR     @SETP1          ; WRITE 500 2WD REC.
053 006317         REWIND                  ; SPACEBACK 500
054 060422         DIA     0,.MTA          ; CHECK CA
055 101203         MOVR    0,0,SNC
056 000776         JMP     .-2
057 020160         LDA     0,D500
060 040244         STA     0,RECTR
061 006267 A50.1:  WRITE
062 000102         102
063 014244         DSZ     RECTR
064 000775         JMP     A50.1
065 006320         BSPNS
066 000764         500.
067 064422 A50.2:  DIA     1,.MTA
070 064422         DIA     1,.MTA
071 064422         DIA     1,.MTA
072 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
073 125203         MOVR    1,1,SNC
074 000773         JMP     A50.2
075 061422         DIB     0,.MTA
076 024104         LDA     1,CRCNT
077 020126         LDA     2,C21           ; SET = 21 FOR ERROR PRINT
000 122415         SUB#    1,0,SNR         ;ADDRESS REG.  WRONG.
001 000403         JMP     .+3             ;SHOULD HAVE COUNTED 500
002 006314         HALT1                   ;REC ON BACKSPACE.
003 000407         JMP     A50E            ;AC1=EXPECTED COUNT, AC0=ACTUAL
004 006272         BSPACE
005 000001         1
006 064422         DIA     1,.MTA
007 020145         LDA     0,C200          ;BACKSPACE 1 DID NOT PRODUCE
010 107415         AND#    0,1,SNR         ;BOT.   ADDR REG COUNTED 500.
011 006300         EHALT                   ;WC REG WAS COUNTING FASTER
012 006277 A50E:   JSR     @LOOP.          ;BAD WC REG.
                ;SPACING CHECK.
                ;DURING SPACING MANY IO COMMANDS ARE
                ;ISSUED TO GENERATE NOISE.

013 006275 A51:    JSR     @SETP1          ;WRITE 500 2WD REC.
014 006317         REWIND                  ;SPACEBACK 500
015 060422         DIA     0,.MTA          ;CHECK CA
016 101203         MOVR    0,0,SNC         ;SPACEBACK 1 AND
017 000776         JMP     .-2             ;CHECK FOR BOT
020 020160         LDA     0,D500
021 040244         STA     0,RECTR
022 006267 A51.1.  WRITE
023 000102         102
024 014244         DSZ     RECTR
025 000775         JMP     A51.1
026 006320         BSPNS                   ;BACKSPACE-NO STALL
027 000764         500.                    ; REC COUNT
030 065422 A51.2:  DIB     1,.MTA
031 065422         DIB     1,.MTA
032 064422         DIA     1,.MTA
033 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
034 125203         MOVR    1,1,SNC
035 000773         JMP     A51.2
```

```
MT110
036 061422          DIB     0,.MTA
037 024104          LDA     1,CRCNT
040 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
041 122415          SUB#    1,0,SNR         ;ADDRESS REG. WRONG.
042 000403          JMP     .+3             ;SHOULD HAVE COUNTED 500
043 006314          HALT1                   ;REC ON BACKSPACE.
044 000407          JMP     A51E            ;AC1=EXPECTED COUNT, AC0=ACTUAL
045 006272          BSPACE
046 000001          1
047 064422          DIA     1,.MTA
050 020145          LDA     0,C200          ;BACKSPACE 1 DID NOT PRODUCE
051 107415          AND#    0,1,SNR         ;BOT.  ADDR REG COUNTED 500.
052 006300          EHALT                   ;WC REG WAS COUNTING FASTER
053 006277 A51E:    JSR     @LOOP.          ;BAD WC REG.
                    ;SPACING CHECK.
                    ;DURING SPACING MANY IO COMMANDS ARE
                    ;ISSUED TO GENERATE NOISE.

054 006275 A52:     JSR     @SETP1          ;WRITE 500 2WD REC.
055 006317          REWIND                  ;SPACEBACK 500
056 060422          DIA     0,.MTA          ;CHECK CA
057 101202          MOVR    0,0,SNC         ;SPACEBACK 1 AND
060 000776          JMP     .-2             ;CHECK FOR BOT
061 020160          LDA     0,D500
062 040244          STA     0,RECTR
063 006267 A52.1:   WRITE
064 000102          102
065 014244          DSZ     RECTR
066 000775          JMP     A52.1
067 006320          BSPNS                   ;BACKSPACE-NO STALL
070 000764          500                     ; REC COUNT
071 065422 A52.2:   DIB     1,.MTA
072 065422          DIB     1,.MTA
073 064422          DIA     1,.MTA
074 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
075 125203          MOVR    1,1,SNC
076 000773          JMP     A52.2
077 061422          DIB     0,.MTA
100 024104          LDA     1,CRCNT
101 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
102 122415          SUB#    1,0,SNR         ;ADDRESS REG. WRONG.
103 000403          JMP     .+3             ;SHOULD HAVE COUNTED 500
104 006314          HALT1                   ;REC ON BACKSPACE.
105 000407          JMP     A52E            ;AC1=EXPECTED COUNT, AC0=ACTUAL
106 006272          BSPACE
107 000001          1
110 064422          DIA     1,.MTA
111 020145          LDA     0,C200          ;BACKSPACE 1 DID NOT PRODUCE
112 107415          AND#    0,1,SNR         ;BOT.  ADDR REG COUNTED 500.
113 006300          EHALT                   ;WC REG WAS COUNTING FASTER
114 006277 A52E:    JSR     @LOOP.          ;BAD WC REG.
                    ;SPACING CHECK.
                    ;DURING SPACING MANY IO COMMANDS ARE
                    ;ISSUED TO GENERATE NOISE.

115 006275 A53:     JSR     @SETP1          ;WRITE 500 2WD REC.
116 006317          REWIND                  ;SPACEBACK 500
117 060422          DIA     0,.MTA          ;CHECK CA
120 101202          MOVR    0,0,SNC         ;SPACEBACK 1 AND
121 000776          JMP     .-2             ;CHECK FOR BOT
```

```
MT110
022 020160          LDA     0,D500
023 040244          STA     0,RECTR
024 006267 A53.1:   WRITE
025 000102          102
026 014244          DSZ     RECTR
027 000775          JMP     A53.1
030 006320          BSPNS                   ;BACKSPACE-NO STALL
031 000764          500.                    ; REC COUNT
032 065422 A53.2:   DIB     1,.MTA
033 065422          DIB     1,.MTA
034 064422          DIA     1,.MTA
035 125203          MOVR    1,1,SNC
036 000774          JMP     A53.2
037 061422          DIB     0,.MTA
040 024104          LDA     1,CRCNT
041 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
042 122415          SUB#    1,0,SNR         ;ADDRESS REG. WRONG.
043 000403          JMP     .+3             ;SHOULD HAVE COUNTED 500
044 006314          HALT1                   ;REC ON BACKSPACE.
045 000407          JMP     A53E            ;AC1=EXPECTED COUNT, AC0=ACTUAL
046 006272          BSPACE
047 000001          1
050 064422          DIA     1,.MTA
051 020145          LDA     0,C200          ;BACKSPACE 1 DID NOT PRODUCE
052 107415          AND#    0,1,SNR         ;BOT.  ADDR REG COUNTED 500.
053 006300          EHALT                   ;WC REG WAS COUNTING FASTER
054 006277 A53E:    JSR     @LOOP.          ;BAD WC REG.
                    ;LONGITUDINAL PARITY CHECK

055 020455          LDA     0,LSTRT         ;CAUSE LONG PARITY ERROR
056 040453          STA     0,LPOIT         ;IN 2 TRK PAIRS
057 102400          SUB     0,0             ;CHECK PARITY STATUS
060 040450          STA     0,RBCT
061 006266          ERASE
062 006270 A54:     GEN
063 037477          37477
064 020063          LDA     0,OBUFF
065 101400          INC     0,0
066 040170          STA     0,IIDX0
067 022442          LDA     0,@LPOIT
070 101005          MOV     0,0,SNR
071 000442          JMP     CKED
072 010170 A54.1:   ISZ     IIDX0           ;%
073 042170          STA     0,@IIDX0
074 102400          SUB     0,0
075 024165          LDA     1,M400
076 010170          ISZ     IIDX0           ;%
077 042170          STA     0,@IIDX0
100 125404          INC     1,1,SZR
101 000775          JMP     .-3
102 006275          JSR     @SETP1
103 020063          LDA     0,OBUFF         ; GET OUTPUT BUFFER ADDRESS
104 062022          DOB     0,.MTA          ; SET MA REGISTER
105 020165          LDA     0,M400          ; GET RECORD LENGTH
106 063222          DOCC    0,.MTA          ; SET WORD COUNT AND CLEAR
107 020144          LDA     0,C150          ; GET EVEN PARITYU WRITE
110 024234          LDA     1,CX            ; GET UNIT NUMBER
111 123000          ADD     1,0             ; CREATE WRITE COMMAND
112 061122          DOAS    0,.MTA          ; START THE WRITE
113 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
```

```
%T110
14 063522          SKPBZ   .MTA            ; TEST FOR WRITE DONE
15 000776          JMP     .-2             ; IF NOT DONE, TEST AGAIN
16 006267          WRITE                   ;NO PARITY ERROR.  # IN AC2
17 000102          102                     ;INDICATES WHICH TRACKS SHOULD
20 125000          MOV     1,1             ;CAUSE ERROR. (SEE BELOW)
21 125000          MOV     1,1             ;0=RB2-P 6=RB2-3  14=RB0-P
22 125000          MOV     1,1             ;1=RB3-P 7=RB3-4  15=RB1-P
                                           ;2=RB4-P 10=RB4-5 16=RB0-1
                                           ;3=RB5-P 11=RB5-6 17=RB0-2
23 000401          JMP     .+1             ;4=RB6-P 12=RB6-7 20=RB1-2
24 006277          JSR     @LOOP.          ;5=RB7-P 13=RB7-2
25 010404          ISZ     LPOIT
26 010402          ISZ     RBCT
27 000733          JMP     A54

30 000000 RBCT:    0
31 000000 LPOIT:   0
32 004533 LSTRT:   LST79
33 060422 CKED:    DIA     0,.MTA
34 024136          LDA     1,C100
35 010774          ISZ     LPOIT
36 123405          AND     1,0,SNR
37 000404          JMP     .+4
40 022771          LDA     0,@LPOIT
41 101004          MOV     0,0,SZR
42 000730          JMP     A54.1
43 006317          REWIND                  ;TEST END
44 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
45 006265          STATUS
46 100201          100201
47 000775          JMP     .-3
50 000401          JMP     .+1

                   ;CHECK FOR DELETE WRITE LOCK

51 020203 ETST:    LDA     0,PASS          ; GET PASS COUNTER
52 101004          MOV     0,0,SZR         ; TEST FOR FIRST PASS
53 000441          JMP     ET.T4           ; IF NOT FIRST, SKIP TESTS
54 022205          LDA     0,@ISWR?EG
55 126520          SUBZL   1,1             ; SET MASK BIT
56 123414          AND#    1,0,SZR
57 000420          JMP     ET.T3
60 006307          JSR     @PINWL
61 006317          REWIND
62 006265          STATUS
63 100205          100205
64 000404          JMP     .+4             ;RING IN
65 000401          JMP     .+1
66 006207          MESSAGE
67 004510          PRB
70 060222          NIOC    .MTA            ; CLEAR CONTROLLER
71 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
72 006317          REWIND
73 006265          STATUS
74 100201          100201
75 000773          JMP     .-5
76 000401          JMP     .+1

                   ;CHECK FOR DELETE EOT TEST
```

```
MT110
77 022205 ET.T3:  LDA     2,@ISWR?EG
00 034105         LDA     3,C2            ; GET MASK BIT
01 173414         AND#    3,2,SZR
02 000402         JMP     .+2
03 006310         JSR     @PETTA
04 006207         MESSAGE
05 004422         MCYC
06 006317         REWIND
07 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
10 006265         STATUS
11 100201         100201
12 000775         JMP     .-3
13 000401         JMP     .+1
14 010203 ET.T4:  ISZ     PAS?S           ; COUNT THE PASS
15 101001         MOV     0,0,SKP         ; SKIP IF COUNTER NOT FULL
16 010203         ISZ     PAS?S           ; SET TO ONE IF FULL
17 022205         LDA     0,@ISWR?EG      ; GET SWREG VALUE
20 024153         LDA     1,C4K           ; GET PASS PRINT BIT
21 107414         AND#    0,1,SZR         ; TEST FOR PRINT PASS
22 002406         JMP     @ET.T5          ; IF SET, NO PRINT
23 006207         MESSAGE
24 003420         MSG2
25 024203         LDA     1,PAS?S         ; GET THE PASS NUMBER
26 006212         TYPED                   ; PRINT IT IN DECIMAL
27 002401         JMP     @.+1

30 000531 ET.T5:  A1
```

```
MT110
                ;MTA DIAGNOSTIC INITIALIZATION

031 054240 INITIAL: STA    3,CNTR          ; SAVE RETURN ADDRESS
032 102400         SUB     0,0             ; CLEAR REGISTER
033 040203         STA     0,PASS          ; INITIALIZE PASS COUNTER
034 006207         MESSAGE                 ;NAME
035 006765         DIRT
036 006207         MESSAGE
037 003360         MSG1                    ;PRINT HALT, MOUNT SCRATCH PACK
040 006222         ASCIN                   ; GETT CONTINUE KEY
041 006211         TYPEA                   ; PRINT THE CHARACTER
042 006207 INIT1:  MESSAGE                 ; PRINT UNIT REQUEST
043 003424         MSGUN
044 006216         OCTIN                   ; GET UNIT NUMBER
045 000775         JMP     INIT1           ; IF ERROR REQUEST AGAIN
046 020114         LDA     0,C10           ; GET MAX NUMBER
047 106432         SUBZ#   0,1,SZC         ; TEST BETWEEN 0 AND 7
050 000772         JMP     INIT1           ; IF NOT, REQUEST AGAIN
051 135000         MOV     1,3             ; SAVE UNIT NUMBER
052 030114         LDA     2,C10           ; GET COMMAND
053 173000         ADD     3,2             ;AC2=UNIT # +10
054 050233         STA     2,C1X           ;STORE UNIT#+10
055 054234         STA     3,CX            ;SAVE UNIT#
056 034131         LDA     3,C40           ; GET COMMAND
057 173000         ADD     3,2             ; AC2 = UNIT # + 50
060 050235         STA     2,C5X           ; STORE UNIT # + 50
061 050236         STA     2,CC5X
062 034124         LDA     3,C20           ; GET COMMAND
063 173000         ADD     3,2             ; SET COMMAND + UNIT
064 050237         STA     2,C7X           ; SAVE ERASE COMMAND
065 006207 INIT6:  MESSAGE                 ; ENTER DEVICE CODE 22 OR 62
066 003435         MSGDC
067 006216         OCTIN                   ; GET OPERATOR RESPONSE
070 000775         JMP     INIT6           ; IF ERROR, REQUEST AGAIN
071 020127         LDA     0,C22           ; GET NORMAL DEVICE CODE
072 106415         SUB#    0,1,SNR         ; TEST FOR NORMAL
073 000404         JMP     INIT7           ; IF NORMAL SAVE IT
074 020133         LDA     0,C62           ; GET SECONDARY CODE
075 106414         SUB#    0,1,SZR         ; TEST FOR SECONDARY
076 000767         JMP     INIT6           ; IF NOT, REQUEST AGAIN
077 040243 INIT7:  STA     0,DEVICE        ; SAVE THE NEW CODE

                ;MODIFY ALL .MTA INSTRUCTIONS

500 030060         LDA     2,FIRST
501 021000 GMOR:   LDA     0,0,2
502 024075         LDA     1,C160077
503 123400         AND     1,0
504 026076         LDA     1,@PCDCM
505 106414         SUB#    0,1,SZR
506 000407         JMP     DEV1            ;NOT MTA
507 021000         LDA     0,0,2
510 024077         LDA     1,C177700
511 123400         AND     1,0
512 024243         LDA     1,DEVICE
513 123000         ADD     1,0
514 041000         STA     0,0,2
515 151400 DEV1:   INC     2,2
516 020061         LDA     0,CEND
517 112414         SUB#    0,2,SZR
```

```
MT110
020 000761         JMP     GMOR
021 006223 INIT8:  CALIB                   ; CALIBRATE THE TIME BASE
022 060214         NIOC    RTC             ; CLEAR THE REAL TIME CLOCK
023 006207         MESSAGE                 ; PRINT SWREG SET MESSAGE
024 003455         MSGS
025 034221 INT10:  LDA     3,IOM?OD        ; GET I/O MOD FLAG
026 175005         MOV     3,3,SNR         ; TEST FOR MOD SET
027 000404         JMP     INT11           ; IF NOT SET, TTY INPUT
030 006221         JSR     @IOM?OD         ; INPUT TO AC0
031 000001         1                       ; INPUT CONTROL FLAG
032 000403         JMP     INT12           ; GO TEST SWREG ENTRY
033 063610 INT11:  SKPDN   TTI             ; TEST FOR TTY READY
034 000777         JMP     .-1             ; IF NOT READY, WAIT
035 006206 INT12:  JSR     @IINP?          ; TEST FOR SWITCH COMMAND
036 020123         LDA     0,CONO          ; GET ASCII CONTROL CHARACTER
037 116415         SUB#    0,3,SNR         ; TEST FOR ODT REQUEST
040 000405         JMP     INT13           ; IF ODT, GO ENTER IT
041 020121         LDA     0,ASCR          ; GET ASCII CR CHARACTER
042 116415         SUB#    0,3,SNR         ; TEST FOR EXIT REQUEST
043 000403         JMP     INIT9           ; IF CR, EXIT SWREG ENTRY
044 000761         JMP     INT10           ; IF NOT CR, GET NEXT ENTRY
045 006226 INT13:  JSR     @IODT?          ; ENTER ODT HERE
046 002240 INIT9:  JMP     @CNTR
                   ;COMPARE IBUFF WITH OBUFF
                   ;
                   ;       CHECK
                   ;       ARG
                   ;
                   ;# WORDS IS IN ARG FOLLOWING CALL
                   ;COMPARE WORD FOR WORD
                   ;WORD N+1 IN IBUFF MUST BE ZERO
                   ;IF ERROR RETURN TO CALL +2
                   ;OTHERWISE CALL+3
                   ;EXIT WITH AC1=BAD WORD
                   ;AC0=GOOD WORD
                   ;AC2=ADDR. OF IBUFF

047 054444 XCHK:   STA     3,RCHK
050 024063         LDA     1,OBUFF
051 044172         STA     1,IIDX2
052 014172         DSZ     IIDX2
053 024064         LDA     1,IBUFF
054 044173         STA     1,IIDX3
055 014173         DSZ     IIDX3
056 031400         LDA     2,0,3           ;GET ARG
057 050240         STA     2,CNTR
060 010172 XCHK.:  ISZ     IIDX2           ;%
061 022172         LDA     0,@IIDX2
062 010173         ISZ     IIDX3           ;%
063 026173         LDA     1,@IIDX3
064 106414         SUB#    0,1,SZR
065 000407         JMP     LEAV
066 014240         DSZ     CNTR
067 000771         JMP     XCHK.
070 010173         ISZ     IIDX3           ;%
071 026173         LDA     1,@IIDX3
072 125005         MOV     1,1,SNR         ;WORD FOLLOWING LAST
                                           ;MUST BE ZERO
073 010420         ISZ     RCHK
074 010417 LEAV:   ISZ     RCHK
```

```
MT110
75 030064          LDA     2,IBUFF
76 002415          JMP     @RCHK

                   ;CLEAR INPUT BUFFER TO ZEROS

77 020064 XCLR:    LDA     0,IBUFF
00 126000          ADC     1,1
01 123000          ADD     1,0
02 040172          STA     0,IIDX2
03 020136          LDA     0,C100
04 040240          STA     0,CNTR
05 102400          SUB     0,0
06 010172          ISZ     IIDX2          ;%
07 042172          STA     0,@IIDX2
10 014240          DSZ     CNTR
11 000775          JMP     .-3
12 001400          JMP     0,3

13 000000 RCHK:    0

                   ;DO A SPACE BACKWARD

14 152401 .XBSP:   SUB     2,2,SKP
15 152520 XBSP:    SUBZL   2,2
16 024131          LDA     1,C40
17 000403          JMP     XSPC0

                   ;DO A SPACE FORWARD

20 152520 XSPC:    SUBZL   2,2
21 024130          LDA     1,C30
22 050433 XSPC0:   STA     2,WSW
23 044431          STA     1,CINS
24 025400          LDA     1,0,3
25 124400          NEG     1,1
26 067022          DOC     1,.MTA         ;SET REC COUNT
27 175400          INC     3,3
30 054423          STA     3,XSPRET       ;RETURN ADDR
31 024136          LDA     1,C100
32 066022          DOB     1,.MTA         ;CA REG = 100
33 024421          LDA     1,CINS
34 030234          LDA     2,CX
35 147000          ADD     2,1
36 065122          DORS    1,.MTA         ;SPACE !!
37 014416          DSZ     WSW
40 002413          JMP     @XSPRET        ;DONT WAIT FOR DONE
41 024125          LDA     1,C20.
42 044245          STA     1,WCTR
43 006224 XSPC1:   WAIT
44 000067          .200MS
45 006313          JSR     @ITEST         ; TEST FOR CONSOLE INPUT
46 063722          SKPDZ   .MTA
47 002404          JMP     @XSPRET        ;DONE
50 014245          DSZ     WCTR
51 000772          JMP     XSPC1
52 002401          JMP     @XSPRET        ;TIMEOUT, 1 SEC
53 000000 XSPRET:  0
54 000000 CINS:    0
55 000000 WSW:     0
```

```
                ;REWIND SUBROUTINE, NO WAIT

56 024233 .RWNS:  LDA     1,C1X
57 065122         DOAS    1,.MTA
60 001400         JMP     0,3

                ;REWIND AND WAIT 20 SEC
                ;OR UNTIL "DONE" AND "TUR"

61 054440 .RW:    STA     3,.RWRET
62 020234         LDA     0,CX
63 061022         DOA     0,.MTA          ;SELECT UNIT
64 020103         LDA     0,C100.
65 040245         STA     0,WCTR
66 060422 RW0:    DIA     0,.MTA          ;READ STATUS
67 034151         LDA     3,C20K
70 163414         AND#    3,0,SZR
71 000413         JMP     .RW2-2          ;REWINDING NOW
72 101202         MOVR    0,0,SZC         ;WAIT FOR "TUR"
73 000407         JMP     RW1             ;TUR=1
74 006224         WAIT
75 000067         .200MS
76 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
77 014245         DSZ     WCTR
00 000766         JMP     RW0
01 002420         JMP     @.RWRET         ;20 SEC. TIMEOUT, NO TUR

02 024233 RW1:    LDA     1,C1X
03 065122         DOAS    1,.MTA          ;REWIND !!
04 024103         LDA     1,C100.
05 044245         STA     1,WCTR
06 006224 .RW2:   WAIT
07 000067         .200MS
10 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
11 064422         DIA     1,.MTA          ;READ STATUS
12 020146         LDA     0,C201          ;TUR + BOT
13 107400         AND     0,1
14 106405         SUB     0,1,SNR
15 002404         JMP     @.RWRET
16 014245         DSZ     WCTR
17 000767         JMP     .RW2
20 002401         JMP     @.RWRET         ;TIMEOUT 20 SEC

21 000000 .RWRET: 0


                ;WRITE EOF
                ;WAIT 1 SEC OR UNTIL DONE

22 054420 XWEOF:  STA     3,RWEOF
23 024234         LDA     1,CX
24 030132         LDA     2,C60
25 147000         ADD     2,1
26 065122         DOAS    1,.MTA
27 020125         LDA     0,C20.
30 040245         STA     0,WCTR
31 006224         WAIT
32 000066         .50MS
33 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
34 063722         SKPDZ   .MTA
```

```
.T110
35 002405          JMP     @RWEOF
36 014245          DSZ     WCTR
37 000772          JMP     .-6
40 002402          JMP     @RWEOF
41 002401          JMP     @RWEOF

42 000000 RWEOF:   0
                   ;LOAD OBUFF WITH 3 WORDS.
                   ;
                   ;       LOAD
                   ;       ARG
                   ;
                   ;ARGUMENT=ADDR OF 3 WORD FIELD

43 031400 XLD:     LDA     2,0,3
44 050173          STA     2,IIDX3
45 030063          LDA     2,OBUFF
46 050174          STA     2,IIDX4
47 014174          DSZ     IIDX4
50 102000          ADC     0,0
51 024161          LDA     1,M3
52 010173          ISZ     IIDX3           ;%
53 032173          LDA     2,@IIDX3
54 113400          AND     0,2
55 010174          ISZ     IIDX4           ;%
56 052174          STA     2,@IIDX4
57 125404          INC     1,1,SZR
60 000772          JMP     .-6
61 175400          INC     3,3
62 001400          JMP     0,3

                   ;GENERATE ARGUMENT DATA INTO OBUFF
                   ;ALWAYS GENERATE 64 WORDS ALL SAME
                   ;
                   ;       GEN
                   ;       ARG
                   ;

63 054416 XGEN:    STA     3,RGEN
64 010415          ISZ     RGEN
65 031400          LDA     2,0,3           ;GET DATA WORD
66 102000          ADC     0,0
67 024063          LDA     1,OBUFF
70 044170          STA     1,IIDX0         ;SET IR0
71 014170          DSZ     IIDX0
72 024164          LDA     1,M100          ;LOAD 64 COUNTER
73 044240          STA     1,CNTR
74 010170          ISZ     IIDX0           ;%
75 052170          STA     2,@IIDX0
76 010240          ISZ     CNTR
77 000775          JMP     .-3
00 002401          JMP     @RGEN

01 000000 RGEN:    0

                   ;SELECT THE UNIT

02 054412 XSEL:    STA     3,RSEL
03 006224          WAIT
04 000065          .25MS
```

```
MT110
05 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
06 020237          LDA     0,C7X
07 061022          DOA     0,.MTA
10 006224          WAIT
11 000065          .25MS
12 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
13 002401          JMP     @RSEL

14 000000 RSEL:    0
                   ;READ SUBROUTINE
                   ;ARGUMENT FOLLOWING CALL SPECIFIES
                   ;#WORDS, AND PARITY TYPE
                   ;BITS 10-15=WORDS       BIT 0=  1 FOR EVEN
                   ;BITS 4-9=RECORDS               0 FOR ODD
                   ;                       BIT 1=  1 FOR TEST MODE

15 054450 XRD:     STA     3,RRD
16 010447          ISZ     RRD
17 031400          LDA     2,0,3
20 020135          LDA     0,C77
21 143400          AND     2,0
22 100400          NEG     0,0
23 040241          STA     0,WDCNT         ;SAVE WORD COUNT
24 102400          SUB     0,0
25 151112          MOVL#   2,2,SZC         ;SKIP IF ODD PAR
26 020136          LDA     0,C100
27 024234          LDA     1,CX
30 123000          ADD     1,0             ;AC0=READ INST.
31 040242          STA     0,INST
32 020154          LDA     0,C7700
33 143400          AND     2,0             ;AC0=#RECX100
34 024064          LDA     1,IBUFF
35 125120          MOVZL   1,1
36 153132          ADDZL#  2,2,SZC
37 125241          MOVOR   1,1,SKP
40 125220          MOVZR   1,1
41 044425          STA     1,CASAV         ;SAVE CA
42 024136          LDA     1,C100          ;AC1=100
43 030241 XRD.1:   LDA     2,WDCNT
44 073022          DOC     2,.MTA          ;SET WC
45 030421          LDA     2,CASAV
46 072022          DOB     2,.MTA          ;LOAD CA REG
47 030242          LDA     2,INST
50 071122          DOAS    2,.MTA          ;START READ
51 030125          LDA     2,C20.
52 050245          STA     2,WCTR
53 006224          WAIT
54 000066          .50MS
55 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
56 063722          SKPDZ   .MTA
57 000403          JMP     .+3
60 014245          DSZ     WCTR
61 000772          JMP     .-6
62 122404          SUB     1,0,SZR
63 000760          JMP     XRD.1
64 002401          JMP     @RRD

65 000000 RRD:     0
66 000000 CASAV:   0
                   ;WRITE NO STALL (WRTNS)
```

```
                ;JUST LIKE (WRITE) EXCEPT THAT ONLY
                ;ONE RECORD IS WRITTEN AND RETURN
                ;IS EXECUTED IMMEDIATELY

57 102000 XWNS:  ADC     0,0
70 040453        STA     0,SWTCH
71 000403        JMP     XWRT+2

                ;WRITE SUBROUTINE
                ;ARGUMENT FOLLOWING CALL SPECIFIES
                ;#RECORDS AND WORDS, AND PARITY.
                ;BITS10-15=#WORDS       BIT 0 = 1FOR EVEN
                ;BITS 4- 9=#RECORDS             0 FOR ODD

72 102400 XWRT:  SUB     0,0
73 040450        STA     0,SWTCH
74 054446        STA     3,RWRT
75 010445        ISZ     RWRT
76 031400        LDA     2,0,3
77 020135        LDA     0,C77
00 113400        AND     0,2
01 150400        NEG     2,2
02 050241        STA     2,WDCNT         ;SAVE WORD COUNT
03 031400        LDA     2,0,3
04 102400        SUB     0,0
05 151112        MOVL#   2,2,SZC         ;SKIP IF ODD PAR
06 020136        LDA     0,C100
07 024235        LDA     1,CSX
10 123000        ADD     1,0
11 040242        STA     0,INST          ;SAVE INST
12 020154        LDA     0,C7700
13 143400        AND     2,0             ;AC0=#RECX100
14 024136        LDA     1,C100
15 030241 XWRT.: LDA     2,WDCNT
16 073222        DOCC    2,.MTA          ;SET WC
17 030063        LDA     2,OBUFF
20 072022        DOB     2,.MTA          ;SET CA
21 030242        LDA     2,INST
22 071122        DOAS    2,.MTA          ;START WRITE
23 030420        LDA     2,SWTCH         ;CHECK SWITCH
24 151004        MOV     2,2,SZR
25 002415        JMP     @RWRT
26 030125        LDA     2,C20.
27 050245        STA     2,WCTR
30 006224        WAIT
31 000066        .50MS
32 006313        JSR     @ITEST          ; TEST FOR CONSOLE INPUT
33 063722        SKPDZ   .MTA
34 000403        JMP     .+3
35 014245        DSZ     WCTR
36 000772        JMP     .-6
37 122404        SUB     1,0,SZR
40 000755        JMP     XWRT.
41 002401        JMP     @RWRT

42 000000 RWRT:  0
43 000000 SWTCH: 0
                ;ERASE ROUTINE

44 020234 XERAS: LDA     0,CX
```

```
MT110
045 054417          STA     3,XERAR
046 024134          LDA     1,C70
047 123000          ADD     1,0
050 061122          DOAS    0,.MTA
051 020125          LDA     0,C20.
052 040245          STA     0,WCTR
053 020124          LDA     0,C20
054 063522          SKPBZ   .MTA
055 000777          JMP     .-1
056 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
057 063722          SKPDZ   .MTA
060 002404          JMP     @XERAR
061 014245          DSZ     WCTR
062 000772          JMP     .-6
063 002401          JMP     @XERAR
064 000000 XERAR:   0

                    ;CHECK STATUS FOR TUR AND ERASE.
                    ;AC0=UNIT BEING SELECTED.
                    ;RETURN+1 IF ERROR

065 054421 CSTAT:   STA     3,CSTAR
066 034234          LDA     3,CX            ; GET SELECTED UNIT NUMBER
067 024113          LDA     1,C7
070 123400          AND     1,0             ;AC0=ACTUAL UNIT BEING SEL.
071 162415          SUB#    3,0,SNR
072 000407          JMP     ON              ;TUR SHOULD BE ON
073 126520          SUBZL   1,1             ;UNIT NOT SEL.
074 060422          DIA     0,.MTA
075 123415          AND#    1,0,SNR         ;CORRECT STATUS = 0
076 010410 OKRET:   ISZ     CSTAR           ;OK, RETURN+2
077 030232          LDA     2,X70           ;X70=UNIT#+70
100 002406          JMP     @CSTAR
101 006265 ON:      STATUS
102 100201          100201
103 000774          JMP     OKRET+1         ;ERROR,AC1=GOOD STAT.
104 101000          MOV     0,0
105 000771          JMP     OKRET           ;AC0=BAD

106 000000 CSTAR:   0
                    ;COMBINE ARGUMENT STATUS WITH SWR
                    ;STATUS AND CHECK AGAINST ACTUAL.
                    ;RETURN+3 IF STATUS MATCH, +1 OTHERWISE
                    ;EXIT WITH AC1=EXPECTED STATUS, AC0=ACTUAL
                    ;
                    ;       STATUS
                    ;       ARG
                    ;

107 054414 XSTAT:   STA     3,STATR
110 010413          ISZ     STATR
111 025400          LDA     1,0,3
112 034136          LDA     3,C100          ;9TRK=1
113 167000          ADD     3,1             ; IF 9 TRACK, SET BIT
114 034153          LDA     3,C4K           ;HI DENS=1
115 167000          ADD     3,1             ;AC1=EXPECTED STATUS
116 060422          DIA     0,.MTA          ;AC0=ACTUAL STATUS
117 034404          LDA     3,STATR
120 106415          SUB#    0,1,SNR
121 001402          JMP     2,3
```

```
MT110
22 001400          JMP     0,3

23 000000 STATR:   0
                   ;END TAPE TEST.  WRITE TO EOT AND
                   ;HALT AFTER MESSAGE.  NOTE ERRORS
                   ;ALONG THE WAY

24 006264 ETT:     JSR     @INTIL
25 004406          JSR     ETTA
26 006317          REWIND                  ; REWIND TO BOT AT END
27 006207          MESSAGE
30 003514          MSGTC
31 006226          JSR     @IODT?          ; END OF TEST MESSAGE
32 000777          JMP     .-1
33 054427 ETTA:    STA     3,ETTB
34 006275          JSR     @SETP1          ; SETUP FOR 1 CYCLE
35 102400          SUB     0,0             ; CLEAR FOR REGISTER
36 062022          DOB     0,.MTA          ; CLEAR MA REGISTER
37 063022          DOC     0,.MTA          ; CLEAR WORD COUNT
40 020235          LDA     0,C5X
41 061122          DOAS    0,.MTA          ;WRITE
42 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
43 063522          SKPBZ   .MTA
44 000776          JMP     .-2
45 064422          DIA     1,.MTA          ;READ STATUS
46 030147          LDA     2,C1000
47 147414          AND#    2,1,SZR
50 000407          JMP     ENDT            ;END TAPE
51 125000 ETT.1:   MOV     1,1             ; TEST THE STATUS WORD
52 125000          MOV     1,1
53 125000          MOV     1,1             ; IF ERROR PROCESS IT
54 125000          MOV     1,1
55 006277          JSR     @LOOP.          ; TEST FOR ERROR LOOP
56 000756          JMP     ETTA+1

57 006207 ENDT:    MESSAGE
60 004414          MENDT
61 002401          JMP     @ETTB
62 000000 ETTB:    0

63 000000 RINWL:   0
64 000000 INWLD:   0
65 060022 CDCM:    060022
                   ;WRITE LOCK TEST

66 006264 WLT:     JSR     @INTIL
67 004405          JSR     INWLT
70 006207          MESSAGE
71 003514          MSGTC
72 006226          JSR     @IODT?
73 000777          JMP     .-1
74 054767 INWLT:   STA     3,RINWL
75 060222          NIOC    .MTA            ; CLEAR CONTROLLER
76 006275          JSR     @SETP1
77 006317          REWIND
300 006265 STA1:   STATUS
301 100205         100205
302 000403         JMP     STA2            ;RING IN OR OTHER TROUBLE
303 101000         MOV     0,0
304 000405         JMP     STA3            ;RING OUT
```

```
MT110
005 006265 STA2:   STATUS
006 100201         100201
007 000771         JMP     STA1            ;RING OUT OR OTHER TROUBLE
010 101000         MOV     0,0
011 006265 STA3:   STATUS                  ;RING IN
012 100201         100201                  ;SKP+2 IF RING IN
013 000422         JMP     INWLA
014 101000         MOV     0,0
015 006207         MESSAGE                 ;RING IN
016 004426         INWLB
017 020157         LDA     0,D3000
020 040744         STA     0,INWLD
021 006265 STA4:   STATUS
022 100201         100201
023 000412         JMP     INWLA           ;GOT RING OUT OR UNRDY
024 101000         MOV     0,0
025 006224         WAIT
026 000136         C100
027 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
030 014734         DSZ     INWLD
031 000770         JMP     STA4
032 006207         MESSAGE                 ;WAITED LONG TIME
033 004471         INWLH
034 002727         JMP     @RINWL
035 020162 INWLA:  LDA     0,M4
036 040726         STA     0,INWLD
037 060222 INWLF:  NIOC    .MTA            ; CLEAR CONTROLLER
040 006317         REWIND
041 006265         STATUS
042 100205         100205
043 000774         JMP     INWLF           ;RING IN OR OTHER
044 101000         MOV     0,0
045 006267         WRITE                   ;OK RING OUT
046 000102         102
047 063522         SKPBZ   .MTA
050 000777         JMP     .-1
051 006265         STATUS
052 110205         110205
053 006300         EHALT                   ;NOT ILLEGAL OR OTHER ERROR
054 000277         LOOP.                   ;AC1-EXPECTED STATUS,AC0-ACTUAL STATUS
055 010707         ISZ     INWLD           ;ILLEGAL
056 000761         JMP     INWLF
057 002704         JMP     @RINWL
    000001         .NOMAC  X
060 106615 MSG1:   .TXTE   !<15><15><12>MOUNT A WRITE ENABLED SCRATCH TAPE, HIT ANY
107 045640 KEY TO CONTINUE!
120 005215 MSG2:   .TXTE   !<15><12>PASS !
124 005215 MSGUN:  .TXTE   !<15><12>DRIVE UNIT #: !
135 005215 MSGDC:  .TXTE   !<15><12>ENTER DEVICE CODE 22 OR 62.  !
155 005215 MSGS:   .TXTE   !<15><12>SET SWITCH REGISTER TO DESIRED VALUE,
201 054724 TYPE CR TO CONTINUE.  !
214 005215 MSGTC:  .TXTE   !<15><12>TEST COMPLETE, WAITING IN ODT.  !
235 005215 MSGWT:  .TXTE   !<15><12>WAITING IN ODT ON ERROR, TYPE A 'P'
260 147724 TO CONTINUE.  !
267 005215 MSGS1:  .TXTE   !<15><12>EXPECTED STATUS = !
302 120254 MSGS2:  .TXTE   !, ACTUAL STATUS = !
314 120254 MSGS3:  .TXTE   !, PC = !
320 005215 MSGLP:  .TXTE   !<15><12>LOOPING ON ERROR!
332 005215 MSGPC:  .TXTE   !<15><12>PERCENT ERROR RATE = !
346 005215 EMG01:  .TXTE   !<15><12>SELD LINE NOT RESET BY IORST!
```

```
MT110
066 005215 EMG02:  .TXTE   !<15><12>BUSY FLIP-FLOP NOT RESET ERROR!
107 005215 EMG03:  .TXTE   !<15><12>BUSY FLIP-FLOP NOT RESET BY IORST!
131 005215 EMG04:  .TXTE   !<15><12>BUSY FLIP-FLOP NOT SET ERROR!
151 005215 EMG05:  .TXTE   !<15><12>DONE FLIP-FLOP NOT RESET ERROR!
172 005215 EMG06:  .TXTE   !<15><12>DONE FLIP-FLOP NOT SET ERROR!
212 005215 EMG07:  .TXTE   !<15><12>SEND CLOCK BIT ON TOO LONG ERROR!
234 005215 EMG10:  .TXTE   !<15><12>FIRST CHARACTER TIME OUT ERROR!
255 005215 EMG11:  .TXTE   !<15><12>DATA TRANSFER TIME OUT ERROR!
275 005215 EMG12:  .TXTE   !<15><12>NO INTERRUPT ERROR!
310 005215 EMG13:  .TXTE   !<15><12>ILLEGAL INTERRUPT WITH MASK BIT SET,
333 046640  MASK = !
340 005215 EMG14:  .TXTE   !<15><12>MTU SELECT ERROR, DIB COMMAND = !
362 005215 EMG15:  .TXTE   !<15><12>MA REGISTER NOT RESET BY IORST!
403 005215 EMG16:  .TXTE   !<15><12>MA REGISTER SETTING ERROR!
421 005215 EMG17:  .TXTE   !<15><12>DATA TRANSFER ERROR, !
435 005215 EMG20:  .TXTE   !<15><12>INTA DEVICE CODE ERROR<15><12>INTA
455 142504 DEVICE CODE = !
465 005215 EMG21:  .TXTE   !<15><12>MA REGISTER COUNTING ERROR<15><12>
504 147507 GOOD VALUE = !
513 005215 MSGCE:  .TXTE   !<15><12>DATA COMPARE ERROR<15><12>!
527 147507 MSGWD:  .TXTE   !GOOD WORD = !
536 120254 MSGBW:  .TXTE   !, BAD WORD = !
545 120254 MSGWA:  .TXTE   !, MEMORY ADDRESS = !
557 120254 MGDOB:  .TXTE   !, DOB = !
564 120254 MGDIB:  .TXTE   !, DIB = !
571 120254 MSGBV:  .TXTE   !, BAD VALUE = !
601 120254 MGUDC:  .TXTE   !, UNIT DEVICE CODE = !
614 005215 MENDT:  .TXTE   !<15><12>END TAPE!
622 005215 MCYC:   .TXTE   !<15><12>CYCLE!
626 005215 INWLB:  .TXTE   !<15><12>MOVE CARTRIDGE SWITCH TO THE SAFE POSITION. DONT STOP THE PROGRAM. !
671 005215 INWLH:  .TXTE   !<15><12>WRITE LOCK BIT OFF: 100201 !
710 005215 PRB:    .TXTE   !<15><12>MOVE SWITCH OFF THE SAFE POSITION !
                   ;LONGITUDINAL PARITY DATA LIST

733 004000 LST79:  4000                 ;RB2-P
734 010000         10000                ;3-P
735 020000         20000                ;4-P
736 000400         400                  ;5-P
737 001000         1000                 ;6-P
740 002000         2000                 ;7-P
741 030000         30000                ;2-3
742 014000         14000                ;3-4
743 006000         6000                 ;4-5
744 003000         3000                 ;5-6
745 001400         1400                 ;6-7
746 000600         600                  ;7-2
747 000000         0


750 100000         100000               ;0-P
751 040000         40000                ;1-P
752 140000         140000               ;0-1
753 120000         120000               ;0-2
754 060000         60000                ;1-2
755 000000         0
756 000000         0

                   ;BAD TAPE STATUS WRITE BUFFER

757 020001 BTBF:   020001               ;1
```

```
MT110
60 010002          010002                  ;2
61 004004          004004                  ;3
62 002010          002010                  ;4
63 000000          000000                  ;5
64 000002          000002                  ;6
65 000420          000420                  ;7
66 020040          020040                  ;8

                   ;CRC CHECK WRITE BUFFERS

67 000200 PAT1:    000200
70 000200          000200
71 153600          153600
72 177474 PAT2:    177474
73 136074          136074
74 112074          112074
75 052652 PAT3:    052652
76 076232          076232
77 037232          037232
00 167141 PAT4:    167141
01 000200          000200
02 063600          063600

03 000077 PAT5:    77
04 000077          77
05 000077          77
06 037400 PAT6:    037400
07 037400          037400
10 037400          037400
11 012452 PAT7:    012452
12 012452          012452
13 012452          012452
14 012452 PAT8:    012452
15 012477          012477
16 012477          012477
17 054420          STA     3,LOOPR
20 176520          SUBZL   3,3
21 000403          JMP     .+3
22 054415 ENTER:   STA     3,LOOPR         ;LOOP ITERATE RETURN
23 034407          LDA     3,ITR           ;THIS ROUTINE INITIALIZES
24 054407          STA     3,ITRCT         ;EACH TEST
25 176400          SUB     3,3
26 054406          STA     3,ESWIT
27 054406          STA     3,ERRCT
30 060222          NIOC    .MTA            ; CONTROLLER RESET
31 002406          JMP     @LOOPR

32 000012 ITR:     12
33 000000 ITRCT:   0
34 000000 ESWIT:   0
35 000000 ERRCT:   0
36 000000 RETURN:  0
37 000000 LOOPR:   0

40 054776 CYCLE:   STA     3,RETURN        ;END OF TEST ITERATION
41 050531          STA     2,SAV2          ;ROUTINE
42 044527          STA     1,SAV1          ;SAVE THE ACS'
43 040525          STA     0,SAV0
44 014767          DSZ     ITRCT
45 000433          JMP     CYCTS           ;NOT 10 TIMES ITERATED
```

```
 46 034764          LDA     3,ITR           ;RESET ITERATION CNTR
 47 054764          STA     3,ITRCT
 50 036205          LDA     3,@ISWR?EG      ; GET SWITCH VALUES
 51 030763          LDA     2,ESWIT         ;IF SWITCH 3=(1)
 52 175120          MOVZL   3,3             ;AND A ERROR HAS OCCURED
 53 175100          MOVL    3,3             ;THE ERROR RATE WILL
 54 151005          MOV     2,2,SNR         ;BE PRINTED
 55 000414          JMP     NOEX
 56 177103          ADDL    3,3,SNC
 57 000417          JMP     PCENT

 60 006207          MESSAGE                 ; PRINT PERCENT MESSAGE
 61 003632          MSGPC
 62 024753          LDA     1,ERRCT
 63 030143          LDA     2,C144
 64 006227          JSR     @IMLT?          ;AC1XAC2
 65 030745          LDA     2,ITR
 66 006230          JSR     @IDVI?          ;AC0-1/AC2
 67 006212          TYPZD                   ;PRINT VALUE
 70 000406          JMP PCENT

 71 020477 NOEX:    LDA     0,SAV0          ;NORMAL EXIT,NO ERR
 72 024477          LDA     1,SAV1
 73 030477          LDA     2,SAV2
 74 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
 75 002741          JMP     @RETURN


 76 102400 PCENT:   SUB     0,0
 77 040736          STA     0,ERRCT         ;RESET ERROR COUNT
100 020470 CYCTS:   LDA     0,SAV0          ;RESTORE ACS
101 024470          LDA     1,SAV1
102 030470          LDA     2,SAV2
103 060222          NIOC    .MTA            ; CLEAR CONTROLLER
104 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
105 034727          LDA     3,ESWIT
106 175005          MOV     3,3,SNR         ; TEST FOR ERROR SWITCH
107 002730          JMP     @LOOPR          ; IF NOT SET, LOOP
110 036205          LDA     3,@ISWR?EG      ; GET SWITCH REGISTER
111 177113          ADDL#   3,3,SNC         ;SWITCH 1
112 002725          JMP     @LOOPR          ;(0)=LOOP ROUTINE
113 002723          JMP     @RETURN         ;(1)=PROCEED TO NEXT TEST

114 054722 SPER1:   STA     3,RETURN        ; SAVE RETURN ADDRESS
115 176520          SUBZL   3,3             ; SET BIT 15
116 000406          JMP     EERR            ; GO SET CONTROL FLAG

117 054717 SPER2:   STA     3,RETURN        ; SAVE RETURN ADDRESS
120 176620          SUBZR   3,3             ; SET BIT 0
121 000403          JMP     EERR            ; GO SET CONTROL FLAG

122 054714 ERR:     STA     3,RETURN        ;ERROR SUBROUTINE
123 176400          SUB     3,3             ; CLEAR REGISTER
124 054473 EERR:    STA     3,EFLAG         ; SET ERROR CONTROL FLAG
125 050445          STA     2,SAV2
126 044443          STA     1,SAV1
127 040441          STA     0,SAV0

130 034704          LDA     3,ESWIT
```

```
MT110
031 175005          MOV     3,3,SNR
032 000413          JMP     ERR1
033 032205          LDA     2,@ISWR?EG      ; GET SWITCH REGISTER
034 024145          LDA     1,C200          ; GET MASK BIT
035 133414          AND#    1,2,SZR         ; TEST FOR PRINT ALL
036 004435          JSR     EPRNT           ; IF SET, PRINT STATUS

037 030433 ERET:    LDA     2,SAV2          ;RESTORE ACS
040 024431          LDA     1,SAV1
041 020427          LDA     0,SAV0
042 010673          ISZ     ERRCT           ;COUNT
043 101000          MOV     0,0             ;ERRORS,I/O RESET
044 002672          JMP     @RETURN         ;EXIT

045 004426 ERR1:    JSR     EPRNT           ; GO PRINT STATUS
046 032205          LDA     2,@ISWR?EG      ; GET SWITCH REGISTER
047 020147          LDA     0,C1000         ; GET MASK BIT
050 113415          AND#    0,2,SNR         ; TEST FOR WAIT ON ERROR
051 000404          JMP     EPRINT          ;PRINT ERROR DATA
052 006207          MESSAGE                 ; PRINT WAIT MESSAGE
053 003535          MSGWT
054 006226          JSR     @IODT?          ; EXIT TO ODT

055 032205 EPRINT:  LDA     2,@ISWR?EG      ; GET SWREG VALUE
056 024156          LDA     1,C40000        ; GET MASK BIT
057 133404          AND     1,2,SZR         ; TEST FOR LOOP ON ERROR
060 000757          JMP     ERET            ; IF NO LOOP, RETURN
061 020655          LDA     0,RETURN        ; GET RETURN ADDRESS
062 040652          STA     0,ESWIT         ; SET THE ERROR SWITCH
063 006207          MESSAGE                 ; PRINT LOOPING MESSAGE
064 003620          MSGLP
065 024645          LDA     1,ITR           ; GET ITERATION COUNT
066 044645          STA     1,ITRCT         ; SET COUNTER
067 000750          JMP     ERET            ; PROCESS RETURN

070 000000 SAV0:    0
071 000000 SAV1:    0
072 000000 SAV2:    0

073 054423 EPRNT:   STA     3,MSAV          ; SAVE THE RETURN
074 034423          LDA     3,EFLAG         ; GET CONTROL FLAG
075 175212          MOVR#   3,3,SZC         ; TEST FOR BIT 15 SET
076 000441          JMP     SPERT           ; IF SET, PROCESS ERROR
077 175112          MOVL#   3,3,SZC         ; TEST FOR BIT 0 SET
100 000420          JMP     CPERT           ; IF SET, PROCESS ERROR
101 006207          MESSAGE                 ; PRINT EXPECTED STATUS
102 003567          MSGS1
103 024766          LDA     1,SAV1          ; GET EXPECTED STATUS
104 006214          TYPEO                   ; PRINT THE STATUS WORD
105 006207          MESSAGE                 ; PRINT ACTUAL STATUS
106 003602          MSGS2
107 024761          LDA     1,SAV0          ; GET THE ACTUAL STATUS
110 006214          TYPEO                   ; PRINT THE STATUS WORD
111 006207 EPRT1:   MESSAGE                 ; PRINT PC
112 003614          MSGS3
113 024623          LDA     1,RETURN        ; GET RETURN ADDRESS
114 006214          TYPEO                   ; PRINT THE PC WORD
115 002401          JMP     @MSAV           ; RETURN TO CALL

116 000000 MSAV:    0
```

```
17 000000 EFLAG:  0

          ; PRINT COMPARE ERROR

20 006207 CPERT:  MESSAGE               ; PRINT ERROR MESSAGE
21 004313         MSGCE
22 006207         MESSAGE               ; PRINT GOOD WORD =
23 004327         MSGWD
24 024744         LDA     1,SAV0        ; GET GOOD WORD
25 006214         TYPEO                 ; PRINT IT IN OCTAL
26 006207         MESSAGE               ; PRINT BAD WORD =
27 004336         MSGBW
30 024741         LDA     1,SAV1        ; GET BAD WORD
31 006214         TYPEO                 ; PRINT IT IN OCTAL
32 006207         MESSAGE               ; PRINT MEMORY ADDRESS =
33 004345         MSGWA
34 024736         LDA     1,SAV2        ; GET MEMORY ADDRESS
35 006214         TYPEO                 ; PRINT IT IN OCTAL
36 000753         JMP     EPRT1         ; GO PRINT PC =

          ; PRINT ALL OTHER ERROR MESSAGES

37 030520 SPERT:  LDA     2,MSGTB       ; GET MESSAGE TABLE ADDRESS
40 024732         LDA     1,SAV2        ; GET MESSAGE NUMBER
41 133000         ADD     1,2           ; SET TO TABLE ADDRESS
42 035000         LDA     3,0,2         ; GET THE MESSWAGE ADDRESS
43 020117         LDA     0,C13         ; GET FOR PRINT CONTROL
44 106432         SUBZ#   0,1,SZC       ; TEST FOR < 13
45 000405         JMP     SPRT1         ; IF NOT < 13, FIND PRINT OUT
46 054402         STA     3,PMSG1       ; SAVE THE MESSAGE ADDRESS
47 006207         MESSAGE               ; PRINT REQUESTED MESSAGE
50 000000 PMSG1:  0                     ; MESSAGE ADDRESS
51 000740         JMP     EPRT1         ; GO PRINT PC =
          ; TEST FOR PRINT INTERRUPT MASK ERROR

52 106414 SPRT1:  SUB#    0,1,SZR       ; TEST FOR MESSAGE 13
53 000407         JMP     SPRT2         ; IF NOT 13, TEST NEXT
54 054402         STA     3,PMSG2       ; SAVE THE MESSAGE ADDRESS
55 006207         MESSAGE               ; PRINT MASK ERROR
56 000000 PMSG2:  0                     ; MESSAGE ADDRESS
57 024712         LDA     1,SAV1        ; GET THE MASK WORD
60 006214         TYPEO                 ; PRINT IT IN OCTAL
61 000730         JMP     EPRT1         ; GO PRINT PC =

          ; TEST FOR PRINT MTU SELECT ERROR

62 101400 SPRT2:  INC     0,0           ; SET = 14
63 106414         SUB#    0,1,SZR       ; TEST FOR MESSAGE 14
64 000407         JMP     SPRT3         ; IF NOT 14, TEST NEXT
65 054402         STA     3,PMSG3       ; SAVE THE MESSAGE ADDRESS
66 006207         MESSAGE               ; PRINT MTU SELECT ERROR
67 000000 PMSG3:  0                     ; MESSAGE ADDRESS
70 024701         LDA     1,SAV1        ; GET DIB COMMAND
71 006214         TYPEO                 ; PRINT IT IN OCTAL
72 000717         JMP     EPRT1         ; GO PRINT PC =

          ; TEST FOR MA REGISTER ERROR

73 020123 SPRT3:  LDA     0,C17         ; SET FOR MESSAGE 17
74 106432         SUBZ#   0,1,SZC       ; TEST FOR < 17
```

```
MT110
75 000415          JMP     SPRT4           ; IF NOT < 17, TEST NEXT
76 054402          STA     3,PMSG4         ; SAVE THE MESSAGE ADDRESS
77 006207          MESSAGE                 ; PRINT THE REQUESTED MESSAGE
00 000000 PMSG4:   0                       ; MESSAGE ADDRESS
01 006207          MESSAGE                 ; PRINT DOB = MESSAGE
02 004357          MGDOB
03 024665          LDA     1,SAV0          ; GET DATA SENT
04 006214          TYPEO                   ; PRINT IT IN OCTAL
05 006207          MESSAGE                 ; PRINT DIB = MESSAGE
06 004364          MGDIB
07 024662          LDA     1,SAV1          ; GET DATA INPUT
10 006214          TYPEO                   ; PRINT IT IN OCTAL
11 000700          JMP     EPRT1           ; GO PRINT PC =

          ; TEST FOR PRINT DATA TRANSFER ERROR

12 106414 SPRT4:   SUB#    0,1,SZR         ; TEST FOR MESSAGE 17
13 000415          JMP     SPRT5           ; IF NOT SET, TEST NEXT
14 054402          STA     3,PMSG5         ; SAVE MESSAGE ADDRESS
15 006207          MESSAGE                 ; PRINT DATA TRANSFER ERROR
16 000000 PMSG5:   0                       ; MESSAGE ADDRESS
17 006207          MESSAGE                 ; PRINT GOOD WORD =
20 004327          MSGWD
21 024650          LDA     1,SAV1          ; GET GOOD DATA WORD
22 006214          TYPEO                   ; PRINT IT IN OCTAL
23 006207          MESSAGE                 ; PRINT BAD WORD =
24 004336          MSGBW
25 024643          LDA     1,SAV0          ; GET BAD DATA WORD
26 006214          TYPEO                   ; PRINT IT IN OCATAL
27 000662          JMP     EPRT1           ; GO PRINT PC =

          ; TEST FOR PRINT INTA DEVICE CODE ERROR

30 101400 SPRT5:   INC 0,0                 ; SET FOR MESSAGE 20
31 106414          SUB#    0,1,SZR         ; TEST FOR MESSAGE 20
32 000413          JMP     SPRT6           ; IF NOT 20, DO 21
33 054402          STA     3,PMSG6         ; SAVE MESSAGE ADDRESS
34 006207          MESSAGE                 ; PRINT INTA DEVICE CODE ERROR
35 000000 PMSG6:   0                       ; MESSAGE ADDRESS
36 024632          LDA     1,SAV0          ; GET INTA DEVICE CODE
37 006215          TYPZO                   ; PRINT IT IN OCTAL
40 006207          MESSAGE                 ; PRINT UNIT DEVICE CODE =
41 004401          MGUDC
42 024627          LDA     1,SAV1          ; GET DEVICE CODE
43 006215          TYPZO                   ; PRINT IT IN OCTAL
44 000645          JMP     EPRT1           ; GO PRINT PC =

          ; PRINT MA REGISTER COUNTING ERROR

45 054402 SPRT6:   STA     3,PMSG7         ; SAVE THE MESSAGE ADDRESS
46 006207          MESSAGE                 ; PRINT MA COUNTING ERROR
47 000000 PMSG7:   0                       ; MESSAGE ADDRESS
50 024621          LDA     1,SAV1          ; GET GOOD VALUE
51 006214          TYPEO                   ; PRINT IT IN OCTAL
52 006207          MESSAGE                 ; PRINT BAD VALUE =
53 004371          MSGBV
54 024614          LDA     1,SAV0          ; GET BAD VALUE WORD
55 006214          TYPEO                   ; PRINT IT IN OCTAL
56 000633          JMP     EPRT1           ; GO PRINT PC =
```

```
MT110
        ; MESSAGE TABLE

057 005157 MSGTB:  .
060 003646         EMG01
061 003666         EMG02
062 003707         EMG03
063 003731         EMG04
064 003751         EMG05
065 003772         EMG06
066 004012         EMG07
067 004034         EMG10
070 004055         EMG11
071 004075         EMG12
072 004110         EMG13
073 004140         EMG14
074 004162         EMG15
075 004203         EMG16
076 004221         EMG17
077 004235         EMG20
000 004265         EMG21

001 000000         0                       ; RETURN ADDRESS
002 054777 TSTIN:  STA     3,TSTIN-1       ; SAVE THE RETURN ADDRESS
003 040422         STA     0,TSAV0         ; SAVE REGISTER 0
004 034221         LDA     3,IOM?OD        ; GET I/O MOD
005 175005         MOV     3,3,SNR         ; TEST FOR FLAG SET
006 000405         JMP     TSTI1           ; IF NOT SET, GET CONSOLE
007 006221         JSR     @IOM?OD         ; GET I/O INPUT
010 000000         0                       ; INPUT REQUEST
011 000404         JMP     TSTI2           ; IF INPUT READY, TEST IT
012 000411         JMP     TSTI3           ; NOT READY, EXIT
013 063610 TSTI1:  SKPDN   TTI             ; TEST FOR TTY READY
014 000407         JMP     TSTI3           ; NOT READY, EXIT
015 006206 TSTI2:  JSR     @IINP?          ; GO TEST FOR SW REG
016 020123         LDA     0,CONO          ; GET CONTROL O CHARACTER
017 116400         SUB     0,3             ; TEST FOR ODT ENTRY
020 020405         LDA     0,TSAV0         ; RESTORE REGISTER 0
021 175005         MOV     3,3,SNR         ; TEST FOR ODT ENTRY
022 006226         JSR     @IODT?          ; IF YES, ENTER ODT
023 020402 TSTI3:  LDA     0,TSAV0         ; RESTORE REGISTER 0
024 002755         JMP     @TSTIN-1        ; RETURN TO CALL

025 000000 TSAV0:  0

                   T?TY0
                   S?WPK

                   O?DTP
                   O?DTD   2
                   T?MER
```

```
MT110
                N?MOV
57 000000 EGGS:   0
60 000000         0
61 000000         0
62 000000         0
63 000000         0
64 000000 SWREG:  0
65 005215 DIRT:   .TXTE   !<15><12>..C.S.I.  MT110D DIAG.(KENNEDY 1/4") REV. 00 !
   027056
   141456
   051456
   144456
   120056
   046640
   130724
   030261
   120104
   144504
   043501
   024056
   142513
   047116
   042305
   120131
   127661
   021264
   120251
   142722
   027126
   030240
   120060
   000000
016 000000 LAST:   0
   000525 RES?T=A0
                  .END
00 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

MT110

000525	12/31	50/34				
000531	12/38	32/27				
000761	15/51					
000773	16/04					
001004	16/10	16/13				
001007	16/12	16/16				
001010	16/20					
001020	16/31					
001030	16/42					
001045	17/02					
001055	17/13					
001065	17/18	17/21				
001070	17/20	17/24				
001032	16/44	16/58				
001071	17/28					
001105	17/34	17/41				
001106	17/45					
000541	12/49					
001117	17/58					
001146	18/07	18/17	18/23			
001121	17/60					
001147	18/27					
001157	18/38					
001170	18/50					
001204	19/05					
001237	19/19	19/29	19/35			
001210	19/09	19/44				
001212	19/06	19/12	19/43			
001250	19/40	19/48				
001303	20/02	20/11	20/17			
001254	19/52	20/26				
001256	19/49	19/55	20/25			
001314	20/22	20/30				
001333	20/48					
001352	21/09					
001344	20/60					
001361	21/18					
001400	21/28	21/33				
000546	12/57					
001401	21/37					
001424	21/42	21/51	21/57			
001425	21/60					
001437	22/15					
001463	22/23	22/29	22/36			
001464	22/41					
001537	22/54	22/59	23/07	23/11	23/21	23/26
001470	22/45	23/35				
001473	22/42	22/48	23/34			
001550	23/31	23/40				
001623	23/53	23/58	24/06	24/10	24/19	24/24
001554	23/44	24/33				
001557	23/41	23/47	24/32			
001634	24/29	24/37				
001676	24/46	24/54	24/58	25/08	25/13	
001677	25/16					
001741	25/25	25/33	25/37	25/48	25/52	
001742	25/53					
001772	26/01	26/10	26/17			
000553	13/05					

MT110

001773	26/21						
002021	26/29	26/37	26/45				
002022	26/48						
002051	26/56	27/05	27/11				
000565	13/18						
002052	28/05						
002112	28/30	28/37					
002061	28/12	28/15					
002067	28/18	28/23					
002113	28/42						
002153	29/07	29/14					
002122	28/49	28/52					
002130	28/55	28/60					
002154	29/19						
002214	29/44	29/51					
002163	29/26	29/29					
002171	29/32	29/37					
002215	29/56						
002254	30/20	30/27					
002224	30/03	30/06					
002232	30/09	30/13					
002262	30/35	31/14					
002272	30/43	31/26					
000575	13/29						
000577	13/31	13/50					
000602	13/30	13/34	13/41	13/49			
000610	13/42	13/48					
000621	13/45	13/54	14/04				
000635	14/05	14/09					
000650	14/23						
000662	14/34						
000674	14/45						
000706	14/56						
000720	15/09						
000736	15/26						
000745	15/36						
000755	15/41	15/44					
000760	15/43	15/47					
000274	10/11						
006222	10/57	33/10					
000166	8/43						
000121	11/04	34/18					
000167	8/44						
006560	50/01						
000500	8/59	12/03					
000202	8/57	8/59					
000006	7/13	12/33					
006561	50/01						
006272	10/36	20/50	21/22	21/46	22/04	26/32	28/31
	29/08	29/45	30/21				
006320	10/53	28/16	28/53	29/30	30/07		
004557	10/11	43/60					
000105	7/50	20/31					
000114	7/57	19/22	20/04	23/14	24/12	25/01	25/41
	33/16	33/20					
000136	8/15	16/56	31/20	35/10	35/39	38/26	38/38
	39/27	39/33	40/53	42/18			
000147	8/25	41/26	46/17				

MT110

000152	8/28	26/05					
000103	8/16	36/13	36/30				
000137	8/17	22/30					
000140	8/18	21/53					
000141	8/19	27/06					
000142	8/20	26/39					
000071	7/37	14/25	15/10				
005665	49/46	49/47					
000115	7/58						
000100	7/44	20/40					
000116	7/59	18/30					
000101	7/45	13/07					
000117	7/60	18/55	47/27				
000120	8/01	13/35					
000143	8/21	45/16					
000121	8/02	11/04	14/15				
000144	8/22	30/56					
000122	8/03	13/06	13/19	13/58	14/29	14/40	14/51
	15/02	15/19					
000075	7/41	33/47					
000123	8/04	11/05	47/59				
000077	7/43	33/53					
000233	9/30	16/23	33/22	36/03	36/28		
000106	7/51	12/50	16/13	16/22	17/16	18/10	32/02
000124	8/05	18/41	33/28	40/07			
000145	8/23	28/34	29/11	29/48	30/24	46/04	
000150	8/26	8/33					
000146	8/24	36/36					
000151	8/27	17/48	36/16				
000125	8/06	35/47	36/55	38/45	39/43	40/05	
000126	8/07	18/13	19/24	20/06	21/54	22/31	23/17
	24/15	25/04	25/44	26/14	26/40	27/07	28/26
	29/03	29/40	30/16				
000072	7/38	14/36	15/11				
000127	8/08	33/35					
000150	8/33						
000107	7/52	15/57					
000130	8/09	35/31					
000155	8/31						
000110	7/53	15/44					
000131	8/10	18/53	33/24	35/25			
000156	8/32	46/25					
000073	7/39	14/47	15/12				
000074	7/40	14/58	15/13				
000153	8/29	32/18	40/55				
000111	7/54	12/58	15/38	16/08			
000102	7/46	13/20	13/32				
000235	9/32	33/26	39/28	41/20			
000112	7/55	17/21	18/03	19/15	19/58	21/02	21/24
	23/03	24/02	24/50	25/29			
000132	8/11	36/52					
000133	8/12	33/38					
000113	7/56	40/24					
000134	8/13	16/42	40/02				
000135	8/14	12/13	13/43	38/20	39/20		
000154	8/30	38/30	39/31				
000237	9/34	33/30	38/02				
005252	49/46						

MT110

006223	10/59	34/02					
006552	50/01						
006553	50/01						
006554	50/01						
006555	50/01						
006463	50/01						
006420	9/19	50/01					
006437	50/01						
006475	50/01						
006470	50/01						
006556	50/01						
003066	38/37	38/41	38/59				
000236	9/33	33/27					
003265	7/42	41/43					
000061	7/29	33/59					
005265	49/46						
005253	49/46						
006303	10/45	23/22	24/20	25/09	25/49		
005271	49/46						
005251	49/46						
005400	49/46						
002654	35/33	35/41	35/58				
002333	30/42	31/19					
006302	10/44	22/60	23/59				
000062	7/30	13/29					
000240	9/35	33/03	34/23	34/46	34/53	35/11	35/15
	37/47	37/50					
000123	11/05	34/15	49/35				
005020	46/45	47/05					
000104	7/48	28/25	29/02	29/39	30/15		
005275	9/08	49/46					
003206	40/22	40/31	40/33	40/40			
003165	10/02	40/22					
000234	9/31	20/30	30/57	33/23	35/42	36/11	36/51
	38/27	39/60	40/23				
004640	10/14	44/55					
004700	44/60	45/32					
000157	8/35	42/11					
000160	8/36	28/10	28/47	29/24	30/01		
005342	49/46						
005343	49/46						
006630	50/01						
006624	9/20	50/01					
005403	49/46						
002515	33/51	33/58					
000243	9/38	13/46	18/43	33/41	33/55		
000503	7/28	12/03	12/07				
000520	12/11	12/16	12/20				
006765	33/07	50/08					
006735	50/02						
006736	50/02						
006753	50/02						
006744	50/02						
006741	9/24	50/02					
006756	50/02						
006755	50/02						
006557	50/01						
000103	7/47	8/16					

006640 50/01
006641 50/01
000200 7/06 7/14 8/57
004724 45/47 45/51 45/55
005017 45/55 46/41 47/01
006757 7/21 50/02
006300 10/40 15/31 16/36 16/48 17/07 17/33 17/39
17/51 18/20 19/33 20/14 20/43 20/54 21/13
21/31 21/41 21/50 22/08 22/22 22/28 22/53
22/58 23/10 23/52 23/57 24/09 24/45 24/57
25/24 25/36 25/60 26/09 26/28 26/36 26/55
27/04 28/36 29/13 29/50 30/26 42/39
003646 42/60 49/04
003666 43/01 49/05
003707 43/02 49/06
003731 43/03 49/07
003751 43/04 49/08
003772 43/05 49/09
004012 43/06 49/10
004034 43/07 49/11
004055 43/08 49/12
004075 43/09 49/13
004110 43/10 49/14
004140 43/12 49/15
004162 43/13 49/16
004203 43/14 49/17
004221 43/15 49/18
004235 43/16 49/19
004265 43/18 49/20
003257 41/28 41/36
004622 10/12 10/13 44/39
004755 46/19 46/24
004773 46/06 46/15 46/40
005011 46/54 47/19 47/33 47/43 47/55 48/13 48/30
48/46 48/59
000300 10/15 10/40
006266 10/32 17/04 17/15 17/30 17/46 30/34
004737 46/08 46/27 46/34
004722 10/15 45/53
004745 46/02 46/15
004635 44/44 44/51 45/15 45/31 46/11
004634 44/43 44/50 45/05 45/37 45/60 46/29
002351 31/36
003224 10/25 41/08
003233 10/23 41/09 41/15 41/34
003262 41/15 41/38 41/39
003251 41/29
002377 31/42 32/01
002414 31/38 32/14
002430 32/20 32/27
000060 7/28 33/45
005560 49/46
006270 10/34 17/59 19/11 19/54 22/47 23/46 24/39
25/18 30/35
002501 33/46 34/01
006314 10/41 12/44 12/52 12/60 13/13 13/24 13/37
13/60 14/18 14/31 14/42 14/53 15/04 15/21
15/42 15/46 15/59 16/11 16/15 16/26 17/19

	17/23	18/06	18/16	18/33	18/45	18/59	19/18
	19/28	20/01	20/10	21/04	21/27	21/56	22/34
	23/06	23/20	24/05	24/18	24/53	25/07	25/32
	25/47	26/16	26/43	27/10	28/29	29/06	29/43
	30/19						
006315	10/42	23/24	24/22	25/11	25/51		
006562	50/01						
006155	49/47	49/50					
000272	10/09	10/36					
000064	7/32	23/13	24/11	24/60	25/40	26/11	34/42
	35/01	35/06	38/32				
000223	9/19	10/59					
000303	10/18	10/45					
000263	10/02	16/47					
000302	10/17	10/44					
000210	9/08	49/47	49/50				
000224	9/20	10/38	50/01				
000230	9/24	45/19	50/01				
000266	10/05	10/32					
000312	10/25	12/05					
000270	10/07	10/34					
000170	8/45	30/39	30/43	30/44	30/47	30/48	37/44
	37/45	37/48	37/49				
000171	8/46	19/08	19/37	19/41	19/42	19/51	20/19
	20/23	20/24	22/44	23/28	23/32	23/33	23/43
	24/26	24/30	24/31				
000172	8/47	34/40	34/41	34/47	34/48	35/09	35/13
	35/14						
000173	8/48	34/43	34/44	34/49	34/50	34/55	34/56
	37/16	37/22	37/23				
000174	8/49	37/18	37/19	37/25	37/26		
000206	9/06	34/14	49/34				
006206	49/50						
000304	10/19	10/46					
000207	9/07	10/54	49/50	50/01			
000227	9/23	45/17	50/01				
005755	49/47						
006042	49/47						
005736	49/47						
006157	49/47	49/50					
005735	49/47						
006160	49/47	49/50					
006047	49/47						
005724	49/47						
005734	49/47						
006110	49/47						
005774	49/47						
006010	49/47						
006156	49/47	49/50					
006152	49/47	49/50					
006153	49/47	49/50					
002442	33/12	33/15	33/18				
002465	33/31	33/34	33/40				
002477	33/37	33/41					
002521	34/02						
002546	34/20	34/23					
002431	10/03	33/03					
005740	49/47						

MT110

006064	49/47						
005742	49/47						
005745	49/47						
005750	9/06	49/47					
006400	49/50						
006142	49/47	49/50					
006150	49/47	49/50					
006012	49/47	49/50					
006411	49/50						
006410	49/50						
006412	49/50						
000242	9/37	38/29	38/43	39/30	39/38		
006154	49/47						
006144	49/47						
006145	49/47						
006146	49/47						
006147	49/47	49/50					
006143	49/47	49/50					
006124	49/47	49/50					
006205	49/50						
002525	34/06	34/21					
002533	34/08	34/12					
002535	34/11	34/14					
002545	34/17	34/22					
000264	10/03	12/34	41/08	41/46			
000010	7/05	7/15					
005630	49/46	49/47					
005737	49/47						
003335	42/07	42/15	42/25				
004426	42/10	43/30					
003264	41/42	42/12	42/20	42/26	42/41		
003337	42/27	42/31	42/42				
004471	42/23	43/31					
003274	10/22	41/47	41/52				
005741	49/47						
000226	7/17	8/58	9/22	34/22	41/13	41/50	46/22
	49/39	49/50					
000221	9/17	12/20	34/06	34/09	49/25	49/28	49/46
	49/47						
000231	9/25	12/25					
000213	9/11	49/47	49/50				
000212	9/10	10/60	11/02				
000214	9/12	10/58	49/50				
000301	10/16	10/43					
000306	10/21	10/47					
000305	10/20	10/39					
000314	10/27	10/41					
000315	10/28	10/42					
000265	10/04	10/31					
000205	9/05	12/27	31/39	32/01	32/17	45/04	45/40
	46/03	46/16	46/24	49/46	49/47		
000313	10/26	28/21	28/58	29/35	30/60	31/28	31/52
	32/09	35/51	36/23	36/34	36/59	38/01	38/06
	38/49	39/47	40/10	41/22	42/19	45/26	45/36
000225	9/21						
000222	9/18	10/57					
000217	9/15	50/01					
000216	9/14	10/56					

MT110

000220	9/16	49/47	49/50				
004632	44/40	44/48	45/02	45/18	46/32		
004633	44/41	44/49	44/59	45/03	46/33		
000211	9/09	11/01	49/46	49/47			
000271	10/08	10/35					
000311	10/24	12/04					
000273	10/10	10/37					
000267	10/06	10/33					
000215	9/13	10/55	49/50				
005304	49/46						
005305	49/46						
007016	7/29	7/31	7/32	50/33			
006563	50/01						
002574	34/52	34/60					
006304	10/46						
004637	44/36	44/39	44/46	44/53	45/39	45/42	
000277	10/14	12/45	12/53	13/01	13/14	13/25	13/40
	14/01	14/19	14/32	14/43	14/54	15/05	15/22
	15/32	15/47	15/60	16/16	16/27	16/38	16/52
	17/09	17/24	17/41	17/53	18/23	18/34	18/46
	19/01	19/35	20/17	20/45	20/56	21/05	21/15
	21/33	21/57	22/11	22/36	23/26	24/24	25/13
	25/52	26/17	26/45	27/11	28/37	29/14	29/51
	30/27	31/11	41/33	42/40			
002331	30/31	30/40	31/12	31/17	31/21	31/24	
000254	9/50	19/38	23/29				
000262	9/59	20/20	24/27				
004533	31/18	43/35					
002332	30/30	31/18					
000164	8/41	37/46					
000161	8/38	37/21					
000162	8/39	42/25					
000165	8/42	30/46	30/54				
000163	8/40						
006566	50/01						
006603	50/01						
004422	32/07	43/29					
004414	41/37	43/28					
006207	10/54	31/49	32/06	32/21	33/06	33/08	33/12
	33/31	34/04	41/11	41/36	41/48	42/09	42/22
	45/13	46/20	46/30	46/46	46/50	46/54	47/05
	47/07	47/11	47/15	47/31	47/39	47/51	48/03
	48/05	48/09	48/20	48/22	48/26	48/38	48/42
	48/51	48/55					
005234	49/46						
005226	9/07	49/46					
004364	43/25	48/10					
004357	43/24	48/06					
004401	43/27	48/43					
005016	46/40	46/58	46/60				
003360	33/09	42/45					
003420	32/22	42/47					
004371	43/26	48/56					
004336	43/22	47/12	48/27				
004313	43/20	47/06					
003435	33/32	42/49					
003620	42/58	46/31					
003632	42/59	45/14					

003455		34/05	42/50					
003567		42/55	46/47					
003602		42/56	46/51					
003614		42/57	46/55					
005157		47/23	49/03					
003514		41/12	41/49	42/52				
003424		33/13	42/48					
004345		43/23	47/16					
004327		43/21	47/08	48/23				
003535		42/53	46/21					
006734		50/02						
006722		50/02						
006733		50/02						
006724		50/02						
006721		9/23	50/02					
004660	MC	50/01						
000045		7/21	12/08					
004671		45/09	45/23					
030316	MC	50/02						
030177	MC	50/01						
030207	MC	50/02						
006162		49/50						
000063		7/31	18/09	19/21	20/03	30/37	30/52	34/39
		37/17	37/43	39/36				
006216		10/56	33/14	33/33				
006157		49/50						
006160		49/50						
006156		49/50						
006364		49/50						
006163		49/50						
006161		49/50						
006306		49/50						
006347		49/50						
006226		49/50						
006223		49/50						
006330		49/50						
006370		49/50						
006353		49/50						
006355		49/50						
006362		49/50						
006366		49/50						
006373		49/50						
006365		49/50						
006274		49/50						
006372		49/50						
006314		49/50						
006371		49/50						
000204		9/01	49/50					
006172		49/50						
006173		49/50						
006217		49/50						
006242		49/50						
006376		49/50						
006207		49/50						
006212		9/22	49/50					
006413		49/50						
006375		49/50						
006367		49/50						

MT110

006245		49/50						
003176		40/31	40/36	40/38				
003201		40/27	40/34					
000527	MC	5/51	49/49					
022740	MC	49/48						
005561		49/46	49/47					
005247		49/46						
005376		49/46						
005523		49/46						
005377		49/46						
000203		8/60	31/36	32/14	32/16	32/23	33/05	
004567		44/11						
004572		44/14						
004575		44/17						
004600		44/20						
004603		44/24						
004606		44/27						
004611		44/30						
004614		44/33						
005667		49/46						
005666		49/46						
005402		49/46						
005250		49/46						
005661		49/46	49/47					
005662		49/46	49/47					
005532		49/46						
005401		49/46						
005533		9/25	49/46					
000076		7/42	33/49					
004676		45/11	45/21	45/30				
005375		49/46						
005335		49/46						
005333		49/46						
005320		9/11	49/46					
005330		9/10	49/46					
000310		10/23	32/05					
000307		10/22	31/43					
005460		49/46						
005050		47/30	47/32					
005056		47/38	47/40					
005067		47/50	47/52					
005100		48/02	48/04					
005116		48/19	48/21					
005135		48/37	48/39					
005147		48/50	48/52					
005312		9/12	49/46					
000246		9/44	19/07	22/43				
000255		9/54	19/50	23/42				
004510		31/50	43/32					
005245		49/46						
005723		49/46						
002330		30/33	31/13	31/16				
002613		34/38	34/59	34/60	35/02	35/19		
006301		10/43	23/01	23/60	24/48	25/27	26/03	
000244		9/39	28/11	28/14	28/48	28/51	29/25	29/28
		30/02	30/05					
000525		49/47	50/34					
004636		44/52	44/55	45/27	45/43	45/45	45/49	45/53

	46/13	46/28	46/56				
006317	10/51	16/33	17/03	17/14	17/29	17/36	17/52
	20/49	21/19	21/38	22/01	22/16	22/19	22/46
	22/55	23/45	23/54	24/38	24/47	25/17	25/26
	25/54	26/02	26/22	26/49	26/59	28/06	28/43
	29/20	29/57	31/27	31/44	31/53	32/08	41/10
	41/55	42/28					
003001	37/39	37/40	37/52	37/54			
003263	41/41	41/52	42/24	42/43			
003065	38/17	38/18	38/56	38/58			
003014	37/58	38/07	38/09				
005413	49/46						
005374	49/46						
005707	49/46						
002666	36/15	36/25					
002702	36/20	36/28					
002742	36/50	37/01	37/04	37/05	37/07		
006316	10/49	17/47					
003142	39/17	39/18	39/42	39/54	39/56		
004770	44/58	45/23	45/32	45/58	46/10	46/36	46/52
	47/09	48/07	48/28	48/40	48/57		
004771	44/57	45/24	45/33	45/57	46/09	46/37	46/48
	47/13	47/41	47/53	48/11	48/24	48/44	48/53
004772	44/56	45/25	45/34	45/56	46/08	46/38	47/17
	47/24						
005405	49/46						
006564	50/01						
006306	10/47	15/27	15/37	15/52	16/05	16/21	16/32
000276	10/13	12/38	12/49	12/57	13/05	13/18	14/09
	14/23	14/34	14/45	14/56	15/09	15/26	15/36
	15/51	16/04	16/44	17/58	20/34	21/09	
000275	10/12	13/31	13/55	16/20	16/31	17/02	17/13
	17/28	17/45	18/27	18/38	18/50	19/09	19/52
	20/48	20/60	21/18	21/37	21/60	22/15	22/45
	23/44	24/37	25/16	25/52	26/21	26/48	28/05
	28/42	29/19	29/56	30/51	41/16	41/54	
006305	10/39	22/24	26/60				
004714	10/27	45/45					
004717	10/28	45/49					
005037	46/43	47/23					
005052	47/29	47/36					
005062	47/37	47/47					
005073	47/49	47/59					
005112	48/01	48/17					
005130	48/18	48/34					
005145	48/36	48/50					
005246	49/46						
003300	41/56	42/03					
003305	41/58	42/01					
003311	41/60	42/05					
003321	42/13	42/21					
003223	40/50	40/51	40/58	41/03			
006265	10/31	16/34	17/05	17/31	17/37	18/18	19/31
	20/12	20/52	21/11	21/29	21/39	21/48	22/06
	22/20	22/26	22/51	22/56	23/08	23/50	23/55
	24/07	24/43	24/55	25/22	25/34	25/58	26/07
	26/26	26/34	26/53	27/02	31/29	31/45	31/54
	32/10	40/34	41/56	42/01	42/05	42/13	42/29

: MT110

		42/37						
006151		49/47						
006551		50/01						
006715		50/01						
006716		50/01						
006764		9/05	50/07					
003143		39/06	39/16	39/40	39/57			
000050	MC	3/04	49/47					
021330	MC	49/46						
005525		49/46						
005531		49/46						
006565		50/01						
006717		50/01						
006720		50/01						
006677		50/01						
006711		50/01						
006704		50/01						
006642		9/21	50/01					
005663		49/46						
005664		49/46						
005670		9/18	49/46					
005534		49/46						
005574		9/15	49/46					
005631		49/46						
005642		49/46						
005570		9/14	49/46					
005577		49/46						
005537		49/46						
005604		49/46						
005610		49/46						
005536		49/46						
005600		49/46						
005404		49/46						
005562		49/46	49/50					
006377		49/50						
005500		49/46						
005421		9/16	49/46					
006374		49/50						
005225		49/24	49/37	49/40	49/43			
005541		49/46						
005213		49/27	49/32					
005215		49/30	49/34					
005223		49/31	49/33	49/40				
005202		10/26	49/23	49/41				
005443		49/46						
006211		11/01	33/11					
006212		11/02	32/24					
006214		10/58	46/49	46/53	46/57	47/10	47/14	47/18
		47/42	47/54	48/08	48/12	48/25	48/29	48/54
		48/58						
006212		10/60	45/20					
006215		10/55	48/41	48/45				
005424		9/09	49/46					
005530		49/46						
013434	MC	49/50						
016366	MC	49/45						
006224		10/38	12/40	14/13	15/55	35/49	36/21	36/32
		36/57	37/59	38/04	38/47	39/45	42/17	

000245	9/40	35/48	35/54	36/14	36/24	36/31	36/40
	36/56	37/02	38/46	38/52	39/44	39/50	40/06
	40/13						
000241	9/36	38/23	38/39	39/23	39/34		
006271	10/35	21/01	21/10	25/57	26/25	26/52	
003266	10/24	41/46					
006267	10/33	18/01	18/28	18/39	18/51	19/13	19/56
	21/20	21/44	22/02	22/17	22/49	23/48	24/41
	25/20	25/55	26/23	26/30	26/50	26/57	28/12
	28/49	29/26	30/03	31/03	42/33		
006273	10/37						
002655	35/32	35/45	35/59				
000232	9/29	16/43	16/45	16/54	16/55	40/32	
002615	10/09	35/24					
002547	10/18	34/38					
002560	34/47	34/54					
002577	10/17	35/06					
003164	40/01	40/12	40/15	40/16			
003144	10/05	39/60					
002763	10/07	37/39					
002743	10/19	37/15					
003015	10/16	38/17					
003043	38/39	38/55					
003002	10/21	37/58					
002620	10/20	35/30					
002622	35/26	35/32					
002643	35/49	35/55					
002653	35/38	35/46	35/53	35/56	35/57		
003207	10/04	40/50					
002722	10/08	36/50					
003067	10/10	39/05					
003072	10/06	39/07	39/15				
003115	39/34	39/53					
005306	9/13	49/46					
005315	49/46						
005524	49/46						
000067	7/35	12/41	14/14	15/56	35/50	36/22	36/33
000065	7/33	37/60	38/05				
000070	7/36						
000066	7/34	36/58	38/48	39/46			
000022	9/38	10/30	12/51	12/59	13/08	13/09	13/21
	13/22	13/33	13/56	13/57	14/11	14/16	14/26
	14/27	14/28	14/37	14/38	14/39	14/48	14/49
	14/50	14/59	14/60	15/01	15/14	15/15	15/16
	15/17	15/18	15/29	15/39	15/40	15/45	15/53
	15/58	16/06	16/07	16/09	16/14	16/24	16/25
	16/46	17/17	17/22	17/49	18/04	18/12	19/16
	19/25	19/59	20/07	20/35	20/36	20/37	20/39
	21/03	21/25	21/52	22/32	23/04	23/16	24/03
	24/14	24/51	25/03	25/30	25/43	26/13	26/41
	27/08	28/07	28/18	28/19	28/20	28/24	28/33
	28/44	28/55	28/56	28/57	29/01	29/10	29/21
	29/32	29/33	29/34	29/38	29/47	29/58	30/09
	30/10	30/11	30/14	30/23	30/53	30/55	30/59
	31/01	31/19	31/51	35/36	35/40	35/44	35/52
	36/04	36/12	36/15	36/29	36/35	36/54	36/60
	38/03	38/40	38/42	38/44	38/50	39/35	39/37
	39/39	39/48	40/04	40/08	40/11	40/29	40/57

MT110

	41/18	41/19	41/21	41/23	41/25	41/53	42/27
	42/35	44/45	45/35				
002661	10/50	36/10					
002706	36/18	36/32	36/41				
002656	10/48	36/03					
002721	36/10	36/26	36/39	36/42	36/44		
002614	10/52	35/23					

```
                ;
                ;
                ;
                ;*******************************************************************************
                ;
                ; DECRIPTION: UNIVERSAL MAGNETIC TAPE RELIABILITY
                ;
                ;
                ; CUSTOM SYSTEMS INC, 1982
                ;*******************************************************************************
                        .TITL   UMTR
   000001               X=1
   000001               .NOMAC  X
                ;1.     PROGRAM NAME:   UMTR.SR
                ;
                ;2.     REVISION HISTORY:
                ;
                ;               REV.      DATE
22 000000               .REV     00.0   ;06/07/82
                ;
                ;3.     MACHINE REQUREMENTS
                ;
                ;       3.1     NOVA OR ECLIPSE FAMILY CPU'S
                ;       3.2     MINIMUM OF 16K MEMORY
                ;       3.3     MAG TAPE COUPLER (CONTROLLER) BOARD
                ;       3.4     TELETYPE OR CRT AND CONTROLLER
                ;       3.5     TAPE DRIVE (S)
                ;4.     TEST REQUIREMENTS
                ;
                ;       N/A
                ;
                ;5.     SUMMARY
                ;
                ;       THE TAPE RELIABILITY PROGRAM IS A MAINTENANCE
                ;       PROGRAM INTENDED TO VERIFY THE MAGNETIC TAPE
                ;       SUB-SYTEM OPERATION.
                ;
                ;6.     RESTRICTIONS
                ;
                ;       ONLY THOSE TAPE DRIVES TO BE TESTED ARE TO
                ;       BE ONLINE.  ALL ONLINE DRIVES MUST BE WRITE ENABLED.
```

```
;7.      PROGRAM DESCRIPTION/THEORY OF OPERATION
;
;       7.1     RANDOM RELIABILITY (SA 500)
;
;               THE RANDOM RELIABILITY TEST WRITES RANDOM
;               LENGTH FILES.  EACH FILE CONSISTS OF FROM
;               1 TO 7 RANDOM LENGTH, RANDOM PATTERN REC-
;               ORDS.  THE RANDOM FILES ARE WRITTEN AND
;               READ THE FULL LENGTH OF THE MEDIA.  IF
;               MORE THAN ONE(1) TAPE DRIVE IS AVAILABLE,
;               A UNIQUE RANDOM FILE WILL BE WRITTEN ON EACH
;               UNIT SEQUENTIALILY.  WHEN EACH UNIT'S EOT
;               SENSOR IS DETECTED, ITS ACCUMULATED
;               HISTORY IS PRINTED AND THE UNIT IS COM-
;               MANDED TO REWIND.  ALL WRITE ENABLED,
;               READY TAPE UNITS WILL BE TESTED.  A UNIT
;               CAN BE MADE READY AND WILL BE TESTED AFTER
;               THE TEST HAS BEEN INITIATED.  IF A UNIT
;               BECOMES NOT READY DURING THE TEST, ITS
;               HISTORY WILL BE PRINTED AND THE UNIT
;               WILL BE REMOVED FROM THE AVAILABLE UNITS
;               LIST.  THE TEST WILL CONTINUE UNTIL STOPPED
;               BY THE OPERATOR.
;
;       7.2     INTERCHANGE TEST, WRITE/READ (SA 501)
;
;               THE INTERCHANGE TEST IS USED TO VERIFY THE
;               INTERCHANGABLITY OF THE TAPE UNITS.  THIS
;               TEST GENERATES 200, 2000 WORD RECORDS OF
;               SKEW PATTERNS FOLLOWED BY 200, 2000 WORD
;               RECORDS OF RANDOM DATA.  AFTER ALL THE
;               ONLINE, WRITE ENABLED UNITS HAVE BEEN
;               WRITTEN, THEY ARE ALL READ TO INSURE
;               PROPER WRITTING.  THE OPERATOR THEN INTER-
;               CHANGES THE TAPES AND PERFORMS ANOTHER
;               READ VERIFICATION.  THIS PROCEDURE IS CON-
;               TINUED UNTIL EACH TAPE HAS BEEN READ BY
;               ALL THE UNITS.  AFTER EACH READ, A SUMMARY
;               OF THE ACCUMULATED STATISTICS FOR EACH
;               UNIT IS PRINTED.  AFTER ALL THE UNITS HAVE
;               BEEN READ, A TEST COMPLETE MESSAGE IS
;               PRINTED.  IF THE OPERATOR WISHES TO CON-
;               TINUE THE TEST, TYPING A 'P' CHARACTER
;               WILL REPEAT THE ENTIRE TEST.
;
;       7.3     INTERCHANGE, READ ONLY (SA 502)
;
;               THE READ ONLY INTERCHANGE TEST PROVIDES
;               A MEANS OF TESTING TAPE UNITS WITH PRE-
;               RECORDED TAPES.  THE TAPES MUST BE RECORDED
;               IN THE FORMAT DESCRIBIED BY SECTION 7.2.
;               THE READ OPERATION IS IDENTICAL TO
;               SECTION 7.2.
;       7.4     COMMAND STRING INTERPRETER (SA 504)
;
;               THE COMMAND STRING INTERPRETER PROVIDES
;               A TROUBLE SHOOTING AID TO ISOLATE A
;               FAULT.  THE OPERATOR CAN SELECT ALL POS-
;               IBLE OPERATING MODES BY RESPONDING TO
;               CONSOLE REQUESTS.  ALL NUMBERS MUST BE
```

ENTERED IN OCTAL.

7.4.1 UNIT

UNIT NUMBER AND/OR CARRIAGE RETURN TO USE PREVIOUS COMMAND STRING. IF ONLY A CARRIAGE RETURN IS TYPED, NO OTHER REQUESTS WILL BE MADE AND THE LAST ENTERED COMMAND STRING WILL BE RUN. THE ENTRY IS IN THE RANGE OF 0 TO 7. THE DEFAULT UNIT NUMBER IS 0.

7.4.2 WC (WORD COUNT)

TYPE AN OCTAL NUMBER TO SELECT THE DATA BLOCK SIZE AND/OR A CARRIAGE RETURN TO USE THE PREVIOUS ENTRY. THE DEFAULT VALUE IS THE MAXIMUM BLOCK SIZE. THE ENTRY IS IN THE RANGE OF 2 TO THE MAXIMUM BLOCK SIZE.

7.4.3 DATA

SELECT ONE OF THE FOLLOWING DATA PATTERNS AND/OR A CARRIAGE RETURN TO USE THE PREVIOUS ENTRY. THE DEFAULT PATTERN IS RANDOM.

RAND - RANDOM
ALL1 - ALL ONE'S
ALL0 - ALL ZERO'S
ALT0 - ALTERNATING ZERO/ONE (000377)
ALT1 - ALTERNATING ONE/ZERO (177400)
FLT0 - FLOATING ZERO
FLT1 - FLOATING ONE
SKEW - SKEW
VARIABLE - THE VARIABLE PATTERN IS ENTERED BY THE OPERATOR AS OCTAL CHARACTER STRINGS. UP TO 8, 16 BIT OCTAL NUMBERS CAN BE ENTERED. THE DATA BUFFER IS BUILT BY REPEATING THE ENTERED CHARACTER STRINGS.

7.4.4 PARITY

TYPE 'EVEN' OR 'ODD' AND/OR CARRIAGE RETURN TO SELECT THE PARITY OR USE THE PREVIOUS ENTRY. THE DEFAULT PARITY IS ODD.

7.4.5 COMMAND STRING

THE OPERATOR CAN SELECT THE SUBSYSTEM OPERATION BY TYPING THE DESIRED COMMANDS AND/OR CARRIAGE RETURN. ALL N(NUMBER) ENTRIES MUST

BE IN OCTAL. IF THE COMMAND STRING
EXCEEDS THE LINE LENGTH, TYPE A
LINEFEED TO CONTINUE ON THE NEXT
LINE. THE FOLLOWING IS A LIST OF
AVAILABLE SUB-SYSTEM COMMANDS.

```
RD N    READ N RECORDS
RW      REWIND
SB N    SPACE BACK N RECORDS
SF N    SPACE FORWARD N RECORDS
WT N    WRITE N RECORDS
WE      WRITE END OF FILE MARK
ER      ERASE 3" OF TAPE
RE      READ END OF FILE MARK
LOOP    LOOP BACK TO FIRST COMMAND
*       LOOP TO HERE
LOOP *  LOOP TO *
```

SAMPLE COMMAND STRINGS

```
RW WT 10 SB 10 RD 10 LOOP
```

THE ABOVE COMMAND STRING WILL REWIND,
WRITE 8 RECORDS, SPACE BACK 8 RECORDS,
AND READ 8 RECORDS. THIS TEST WILL
CONTINUE UNTIL STOPPED BY THE OPERATOR.

```
RW,WT 10,WE * RW,SF,10,SB,10,RD,10,RE,
LOOP *
```

THE ABOVE COMMAND STRING WILL REWIND,
WRITE 8 RECORDS, WRITE AN EOF MARK,
AND THEN LOOP ON REWIND, SPACE FORWARD
8 RECORDS, SPACE BACK 8 RECORDS, READ
8 RECORDS AND READ EOF MARK.

NOTE: EITHER A SPACE OR COMMA CAN BE
USED AS AN ARGUEMENT DELIMITER.
IF AN INCORRECT CHARACTER OR
CHARACTERS ARE TYPED, TYPE A RUB-
OUT CHARACTER TO DELETE THE PRE-
VIOUSLY TYPED CHARACTER. THE DELE-
TED CHARACTER WILL BE PRINTED.

WHILE THE COMMAND STRING IS BEING EXECUTED,
TYPE A 'R' CHARACTER TO CAUSE THE PROGRAM
TO RETURN TO THE UNIT PROMPT. THE ESCAPE
KEY WILL CAUSE THE PROGRAM TO RETURN TO THE
COMMAND STRING ENTRY POINT.

7.5 HISTORY RECOVERY (SA 504)

IF THE PROGRAM HAS STOPPED DURING AN OPERATION,
THE ACCUMULATED ERROR AND PASS HISTORY CAN BE
RECOVERED BY THIS PROGRAM. THIS PROGRAM MUST
BE RUN BEFORE ANY OTHER PROGRAM IS RESTARTED.

TO RETRIEVE THE ACCUMULATED ERROR AND PASS
HISTORY WHILE THE RELIABILITY TEST IS RUNNING,
TYPE A SPACE. THIS WILL CAUSE THE ACCUM-
ULATED HISTORIES OF ALL TESTED UNITS TO BE

```
;               PRINTED.
;8.  OPERATING MODES/SWITCH SETTINGS:
;               SWITCH OPTIONS
;               BIT     OCTAL   BINARY  INTERPRETATION
;                       VALUE   VALUE
;
;               2       20000   0       ENABLE PRINT ON CONSOLE
;                               1       INHIBIT PRINT ON CONSOLE
;
;               5       02000   0       INHIBIT LINEPRINTER
;                               1       ENABLE LINEPRINTER
;
;               7       00400   0       ENABLE PRINT PARITY ERRORS
;                               1       INHIBIT PRINT PARITY ERRORS
        $$WPD:  8
;               "ESC"   THIS COMMAND GIVEN WHILE RUNNING THE
;                       ENTERED COMMAND STRING WILL CAUSE THE
;                       PROGRAM TO RESTART AT THE COMMAND STRING
;                       ENTER PROMPT.
;
;9.      OPERATING PROCEDURES/OPERATOR INPUT
;
;       9.1     PROGRAM LOAD
;
;               LOAD THE PROGRAM BY USING THE BINARY LOADER.
;
;       9.2     STARTING ADDRESSES
;
;               SA      PROGRAM FUNCTION
;
;               500     START RELIABILITY TEST
;               501     START INTERCHANGE TEST, WRITE/READ
;               502     START INTERCHANGE TEST, READ ONLY
;               503     START COMMAND STRING INTERPRETER
;               504     DIRECT ENTRY FOR ERROR LOG RECOVERY
;
;       9.3     PROGRAM OPERATION
;
;               9.3.1   INITIALIZATION
;
;                       THE FOLLOWING MESSAGE IS PRINTED REQUESTING
;                       THE SETTING OF THE SOFT SWITCH REGISTER.
;
;       "SET SWITCH REGISTER TO DESIRED VALUE, TYPE CR TO CONTINUE."
;       M
;       0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
;       1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
;
;       0
;
;                       MODIFY THE SWITCH REGISTER SETTING AS
;                       DESCRIBED IN SECTION 8.3, FOLLOWED BY
;                       A CARRIAGE RETURN.  THE FOLLOWING MESSAGES
;                       WILL BE PRINTED.
;
;                       IF  A REAL TIME CLOCK IS NOT PRESENT IN
;                       THE SYSTEM, THE FOLLOWING MESSAGE WILL
;                       BE PRINTED.
;
;       "TTO BAUD RATE = ?. .."
```

```
;
;                       RESPOND TO THE REQUEST BY TYPING THE
;                       CORRECT CONSOLE DEVICE BAUD RATE FOR
;                       I/O TIMING CALIBRATION.  IF THE RESPONSE
;                       IS 110, THE FOLLOWING REQUEST MESSAGE
;                       WILL BE PRINTED.
;       "10 OR 11# BITS/CHAR = ?"
;
;                       RESPOND TO THE REQUEST BY TYPING 10 OR
;                       11.
;
;                       A REQUEST TO CHANGE THE DEVICE CODE IS
;                       MADE AS FOLLOWS:
;
;       "RELIABILITY TEST"
;       "ENTER DEVICE CODE 22 OR 62, FOLLOWED BY CR.   "
;
;                       ANSWER THE REQUEST BY TYPING OCTAL 22
;                       OR 62.  IF ANY OTHER DEVICE CODE IS
;                       SELECTED, THE DEVICE CODE ENTRY PROMPT
;                       WILL BE PRINTED AGAIN.
;
;       "ENTER 0 TO TEST CRC (NRZI ONLY) 1 IF NOT, FOLLOWED BY CR.
;
;       ANSWER 0 IF TAPE DRIVE IS 800 BPI NRZI OTHERWISE
;       ENTER 1. NEXT A REQUEST IS MADE TO DETERMINE THE ERROR RECOVERY
;       SEQUENCE THATIS TO BE USED. THIS IS DETERMINED BY THE TYPE OF OPERATING
;       SYSTEM THE CONTROLLER WILL BE USED IN. THE REQUST IS MADE AS FOLLOWS:
;
;    "ENTER 1 IF CONTROLLER WILL BE RUN IN AOS SYSTEM 0 IF NOT,FOLLOWED BY CR"
;
;               9.3.2   PROGRAM ENTRY
;
;                       WHEN ENTERING THE RELIABILITY PROGRAM,
;                       THE FOLLOWING MESSAGE WILL BE PRINTED
;
;       "MOUNT SCRATCH TAPE(S), ENTER CR.  TO CONTINUE."
;
;                       THE OPERATOR SHOULD MAKE READY ALL TAPE
;                       UNITS TO BE TESTED.  ANY TAPE UNIT THAT
;                       IS ONLINE WILL BE TESTED.  AFTER ALL
;                       UNITS ARE READY, ENTER CR.  ON THE CON-
;                       SOLE TO CONTINUE.
;
;               9.3.3   INTERCHANGE TEST, WRITE/READ
;
;                       ENTRY TO THE INTERCHANGE TEST IS IDENT-
;                       ICAL TO THE RELIABILITY TEST WITH THE
;                       FOLLOWING EXCEPTION.
;
;       "INTERCHANGE TEST(WRITE/READ)"
;                       AFTER THE INITIALIZATION SECTION, THE
;                       FOLLOWING MESSAGE IS PRINTED.
;
;       "MOUNT SCRATCH TAPE(S), ENTER CR.  TO CONTINUE."
;
;                       MAKE READY ALL TAPE UNITS TO BE TESTED
;                       AND ENTER CR.  TO CONTINUE.
;               9.3.4   INTERCHANGE TEST, READ ONLY
;
```

```
;                       ENTRY TO THE INTERCHANGE TEST IS
;                       IDENTICAL TO THE RELIABILITY TEST
;                       WITH THE FOLLOWING EXCEPTION.
;
;       "INTERCHANGE TEST(READ ONLY)"
;
;                       AFTER THE INTIALIZATION SECTION THE
;                       FOLLOWING MESSAGE IS PRINTED.
;
;       "MOUNT PRE-RECORDED TAPE(S), ENTER CR  TO CONTINUE."
;
;                       MOUNT PRE-RECORDED TAPES ON ALL TAPE
;                       UNITS TO BE TESTED AND ENTER CR..
;
;       9.4     COMMAND STRING INTERPRETER
;
;              9.4.1   INITIALIZATION
;
;                       ALL ERROR AND PASS COUNTERS ARE CLEARED
;                       AND THE FOLLOWING REQUEST MESSAGE IS
;                       PRINTED.
;
;       "SET SWITCH REGISTER TO DESIRED VALUE, TYPE CR TO CONTINUE"
;       M
;       0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
;       X   X   X   X   X   X   X   X   X   X   X   X   X   X   X   X
;
;       0
;
;                       NOTE:   THE "X" VALUE INDICATE THE UN-
;                               KNOWN STATE OF THE COMMAND BITS.
;
;                       RESPOND TO THE REQUEST BY SETTING THE
;                       "SWREG" LOCATION AS DESCRIBED BY SECTION
;                       8.3, FOLLOWED BY A CARRIAGE RETURN.
;
;                       THE MEMORY IS SIZED NEXT AND THE TIME
;                       BASE IS CALIBRATED.  IF A REAL TIME CLOCK
;                       IS NOT PRESENT IN THE SYSTEM, THE FOL-
;                       LOWING REQUEST IS PRINTED.
;
;       'TTO BAUD RATE = ?"
;
;                       RESPOND TO THE REQUEST BY TYPING THE
;                       CORRECT CONSOLE DEVICE BAUD RATE.  IF THE
;                       RESPONSE IS 110, THE FOLLOWING REQUEST
;                       MESSAGE WILL BE PRINTED.
;
;       "# BITS/CHAR = ?"
;
;                       RESPOND TO THE REQUEST BY TYPING 10 OR
;                       11
;              9.4.2   PROGRAM ENTRY
;
;                       THE FOLLOWING MESSAGES ARE PRINTED
;                       INDICATING THE ENTRY TO THE COMMAND
;                       STRING INTERPRETER.
;
;       "COMMAND STRING INTERPRETER"
;       "MAXIMUM WORD COUNT = XXXX"
```

```
;
;                       NOTE:   THE MAXIMUM WORD COUNT VALUE
;                               INDICATES THE LARGEST DATA
;                               BUFFER AVAILABLE.
;
;                       THE SUB-SYSTEM DEFAULT VALUES ARE SET
;                       AS FOLLOWS:
;
;                       UNIT    0
;                       WC      SET TO MAXIMUM WORD COUNT
;                       DATA    RANDOM PATTERN
;                       PARITY  ODD
;
;                       WHEN THE "UNIT" PROMPT IS TYPED, REFER
;                       TO SECTION 7.4, FOR PROGRAM OPERATION.
;10.     PROGRAM OUTPUT/ERROR DESCRIPTION
;
;       ALL ERRORS ARE IDENTIFIED, COUNTED AND PRINTED ON
;       THE BASIS OF THE SETTING OF LOCATION "SWREG".
;
;       IF A UNIT GOES NOT READY, AN APPROPRIATE ERROR
;       MESSAGE AND ITS ACCUMULATED STATISTICAL HISTORY
;       IS PRINTED.  IF ONLY ONE(1) UNIT IS BEING TESTED,
;       AN APPROPRIATE MEESAGE WILL BE PRINTED AND THE
;       PROGRAM WILL WAIT FOR OPERATOR INTERVENTION.  IF
;       MORE THAN ONE UNIT IS AVAILABLE, THE TEST PROCESS
;       WILL CONTINUE.
;
;       ALL ERRORS ARE SOFT UNLESS SPECIFIED AS HARD OR
;       FATAL.
;
;       10.1    STATISTICAL HISTORY PRINTOUT
;
;               THE STATISTICAL HISTORY IS PRINTED FOR
;               EACH UNIT WHEN IT REACHES ITS EOT SENSOR.
;               THE STATISTICAL HISTORY FOR ALL TESTED
;               UNITS CAN BE REQUESTED BY TYPING A
;               "SPACE" CHARACTER.  A SAMPLE OF THE
;               PRINTOUT IS AS FOLLOWS:
;
;       "UNIT    0        1"
;       "PAR WR  1        0"
;       "PAR RD  1        1"
;       "PERM WR 1        0"
;       "PERM RD 0        0"
;       "WDS RD  30348    1075827"
;       "WDS WR  31345    1075827"
;       10.2    STATUS WORD
;
;               BIT     DESCRIPTION
;
;               0       ANY ERROR, SET BY BITS 1,3,5,6,7,8,10,14
;
;               1(E)    DATA LATE
;               2       REWINDING
;               3(E)    ILLEGAL COMMAND
;
;               4       HIGH DENSITY
;               5(E)    PARITY ERROR
;               6(E)    EOT MARK SENSED
```

```
;
;               7(E)    EOF MARK SENSED
;               8(E)    BOT MARK SENSED
;               9       9 TRACK TAPE
;
;               10(E)   BAD TAPE
;               11      SEND CLOCK OR ID STATUS
;               12      FIRST CHARACTER OR CORRECTED ERROR
;
;               13      WRITE LOCKOUT
;               14(E)   CRC ERROR OR ODD REC READ
;               15      UNIT READY
        0?DTD 11
;12.    SPECIAL NOTES
;
;       12.1    MEDIA SELECTION
;
;               IT IS IMPORTANT TO SELECT KNOWN GOOD TAPES
;               WHEN PERFORMING THE RELIABILITY TESTS. USING
;               MARGINAL TAPE MEDIA WILL CAUSE SOFT AND HARD
;               ERRORS TO OCCURE. TO VERIFIY THE SUB-SYSTEM
;               RELIABILITY THE TAPE MEDIA SHOULD NOT INFLUENCE
;               THE PASS OR FAIL CRITERIA.
;
;       12.2    DATA ENTRY
;
;               ALL NUMBER ENTRIES MUST BE ON OCTAL. ANY OTHER
;               ENTRY WILL BE CONSIDERED AS AN ALPHA CHARACTER.
;
;13.    RUN TIME
;
;       THE PROGRAM RUN TIME IS DEPENDENT ON THE LENGTH OF THE
;       TAPE MEDIA.
        .EOT
```

```
.MTR
0 TOTAL ERRORS, 00000 PASS 1 ERRORS
```