Model 340

Data Channel Synchronous Controller

Technical Manual

Document Number: 600-250-00
Revision: A
Date: 3/4/86
Serial No.:

NOTICE

ZETACO, Inc. has prepared this Technical Manual for use by ZETACO personnel and its customers to assist in the installation, operation and maintenance of the Model PE-730 Memory Expansion Unit. The information contained herein is the property of ZETACO, Inc. and shall not be reproduced in whole nor in part without prior written approval of ZETACO, Inc.

ZETACO, Inc. makes every effort to produce quality products and documentation, however, the reader must be cautioned that changes may be made to the product not reflected in the supplied documentation. If you find errors or omissions, please notify ZETACO, Inc. to remedy the problem. ZETACO, Inc. shall not be responsible for any damages (including consequential) caused by reliance on this material as presented.

If installation problems arise after you thoroughly review the manual, please contact the ZETACO Customer Support Hotline at (612) 941-9480.

Sector Slip™ and Virtual Mapping™ are tradenames of ZETACO, Inc.

Eclipse®, Nova®, and MV® are registered trademarks of Data General Corporation.

FAST™ is a trademark of Fairchild Camera & Instrument Corporation.

REVISION HISTORY

ECO No.	Date	Description	Pages
0328	6/28/84	New ZETACO Cover	
0523	3/4/86	New Cover	

CUSTOMER SERVICE

Our warranty attests the quality of materials and workmanship in our products. If malfunction does occur, our service personnel will assist in any way possible. If the difficulty can not be eliminated by use of the following service instructions and technical advise is required, please phone the Custom Systems sales department (612-941-9480) giving the serial number, board name, model number, and problem description. You will be placed in contact with the appropriate technical assistance.

PRODUCT RETURN

Pre-return Checkout.

If controller malfunction is suspected, the use of test software is needed to determine if the controller is the problem and what in particular is wrong with the controller. The tests applicable to this board are listed on the next page of the manual. Please run the test sequence before considering product return.

Returned Material Authorization.

Before returning a product to Custom Systems for repair, please ask our sales secretary for a "Returned Material Authorization" number. Each product returned requires a separate RMA number. Use of this number in correspondence and on a tag attached to the product will ensure proper handling and avoid unnecessary delays.

Returned Material Information.

Information concerning the problem description, system configuration, diagnostic program name, revision level, and results, i.e., error program counter number should be included with the returning material. A form is provided for this information on the next page of the manual.

Packaging.

To safeguard your materials during shipment, please use packaging that is adequate to protect it from damage. Mark the box "Delicate Instrument" and indicate the RMA number(s) on the shipping label.

MATERIAL RETURN INFORMATION

All possible effort to test a suspected malfunctioning controller should be made before returning the controller to Custom Systems, Inc. for repair. This will: 1) Determine if in fact the board is defective (many boards returned for repair are not defective, causing the user unnecessary system down-time, paper work, and handling while proper testing would indicate the board is working properly). 2) Increase the speed and accuracy of a product's repair which is often dependent upon a complete understanding of the user checkout test results, problem characteristics, and the user system configuration. Checkout results for the Data Channel Sync. Controller should be obtained by performing the following tests. (Include error program counter #'s and accumulator contents if applicable).

TEST	RESULTS
4015 CCT Diagnostic	

Other tests performed:

Please allow our service department to do the best job possible by answering the following questions thoroughly and returning this sheet with the malfunctioning board.

1. Does the problem appear to be intermittent or heat sensitive? (If yes, explain).

   _____

2. What operating system are you running under? (AOS RDOS, DDOS, DTOS).

   _____

3. Describe the system configuration (i.e., peripherals, I/O controllers, model of computer, etc.

   _____

   _____

4. Has the controller been returned before? _____ Same problem?_____

   To be filled out by CUSTOMER:

   Model #: _____

   Serial #: _____

   RMA #: _____

   Returned by:_____
   (company name)

WARRANTY

The Custom Systems' Model 340 Data Channel Sync Controller is warranted against defects in material or workmanship for a period of one year from the date of shipment. Defective units covered by this warranty shall be returned to Custom Systems prepaid. Upon Custom Systems' confirmation of the defect, the defective parts or the entire unit shall be repaired or replaced and returned to the Purchaser.

SERVICE

Service of Custom Systems' products is provided at our Minneapolis facility. Products returned to the factory are in most cases repaired and shipped within two days. Service at the Purchaser's facility is also available on a time and expenses basis.

Service of the product by a Purchaser or by skilled personnel in the Purchaser's locale following the warranty period is facilitated by the complete package of documentation provided with each unit.

TABLE OF CONTENTS

Section 1.0 General Description

Custom Systems' Data Channel Synchronous Controller Model 340 provides complete full-duplex interfacing between a Data General Nova line computer and a Bell 201, Bell 301 or equivalent synchronous data set (modem). The unit occupies one subassembly slot in any Nova-line computer chassis.

Data transmission and reception is in synchronous serial form at a maximum rate of 50K BPS (bits-per-second), and operation may be half or full duplex over 2 wire dial-up or 4 wire dedicated phone lines. Full modem control capability is provided to accomodate manual or automatic answer data sets.

The controller features automatic line synchronization, parity generation and checking and end-of-transmission detection. Sync and EOT (end of transmission) characters for the receiver are programmable and are determined by the program. The unit accomodates character widths of 6, 7 or 8 data bits, plus an optional parity bit (odd or even), for maximum character widths of 9 bits.

Data transfers between the controller and computer memory are handled automatically by the Nova's direct memory access facility (data channel) after block transfer initialization by the program. Seperate channels are used for transmitted and received data.

Also featured is an internal clock which allows data exchanges to be timed by the controller for use with an externally clocked data set or a data link operated without a data set. Five speeds are available from 2.4K to 38.4K BPS.

All data set interface signals are $\pm$12 volt EIA standard No. RS-232C.

Section 2.0 Installation

This section provides information for installing the Model 340 Data Channel Sync Controller. Inspect the controller board for any in-transit damage. Contact the carrier and Custom Systems if any damage is discovered, specifying the nature and extent of the damage.

Section 2.0 Installation (continued)

Included with the controller board are the following accessories:

1-Back panel-paddle board internal cable
1-Paddle board test plug
1-Paddle board to data set cable (optional)

Section 2.1 Controller Installation

Before installing the controller board in the computer chassis, be sure the switch and jumper configuration on the controller board is set to provide the proper device code, character size, parity selection, EOT character enable/disable selection, and internal clock speed if used (see Section 5.0).

Install the controller board by inserting it into one of the available input/output slots in the computer chassis. Be sure it is seated properly by pressing in the insertion levers until they are flush with the front board edge.

Section 2.2 Cabling

Using the back panel-paddle board cable, position the block connector over the corresponding row of back panel pins for the slot containing the controller board, aligning the arrow on the block directly above pin B69. Press the connector securely over the pins. Mount the paddle board to the external connector mounting bolts, provided on the computer frame, with stand-offs set between it and the back panel or the next paddle board connected. Mount it with the signal tabs away from the back panel and the ground tabs (those which are all connected together) towards the back panel.

Using the paddle board to data set cable, fasten the 50 pin connector to the paddle board. (Tabs on the paddle board are numbered 1 through 50, bottom to top on the side facing away from the back panel.) Secure the cable to the computer frame using ty-wraps. Connect the 25-D connector at the other end of the cable to the data set used and fasten securely with the holding screws.

Section 3.0 Controller Operation

The Model 340 Data Channel Sync Controller consists of two seperate interfaces: the receiver (SCR), and the transmitter (SCT). Each portion connects seperately to the Nova data channel. Device codes for the receiver and transmitter are 40 and 41, respectively. Additional controllers, if installed, use device code pairs 42-43, 44-45.....74-75. In each case, the receiver uses the even code, the transmitter the odd code.

The receiver portion of the circuitry assembles a serial bit stream from the data set into characters which are then stored, 2 characters per word, in the computers memory for subsequent processing. The transmitter portion of the circuitry disassembles a memory word into a bit stream which is sent serially to the data set.

All transfers between the controller and memory are in full words containing two characters right-justified in each byte; e.g., 6-bit characters would be in bits 2-7 and 10-15 of a memory word. The transmitter takes two characters from the appropriate bits of each word from memory and transmits them, first the right byte and then the left byte. The receiver assembles each pair of characters into the appropriate bits of a word, right to left, for storage in memory. Characters are transmitted and received serially with the least significant bit first (i.e., bit 15 and bit 7).

Characters may contain 6, 7 or 8 data bits, plus one parity bit (odd or even), or no parity. Character format for transmitter and receiver are seperately selectable (Section 5.0).

The parity generation logic automatically appends a parity bit following the last (most-significant) bit of each transmitted character, and the parity checking logic automatically strips the parity bit from the received characters and checks for errors.

Data transfer may be in blocks of up to 4096 (octal 10000) words for receiver or transmitter. Data transfer begins with at least 2 sync characters, and may end with an EOT (end of transmission) character. These two characters are defined for the receiver by the program with a DOA, SCR instruction.

Section 3.1 Receiver Operation

To set up the receiver to handle incoming date, the program must: specify a sync character, supply an initial memory address for data words, either supply a specific word count or specify an EOT character and a word count large enough to handle the entire message, and issue a start command.

Setting SCR busy causes the receiver to monitor the incoming bit stream continuously until it successively receives two of the sync characters defined by the program, synchronizing the receiver to the bit stream. The receiver then ignores additional sync characters until some other character is received; it begins to assemble pairs of characters into words for transfer to the memory locations specified by the address counter. Since reception is serial, the data channel has one-half bit time in which to respond to a request before information is lost; if the channel is late, "TIMING ERROR" sets, but reception continues. If the receiver is so configured, "PARITY ERROR" sets if a character with incorrect parity is received, and reception continues.

If the controller is set for EOT-found enable, and the EOT character defined by the program appears in the input, the receiver accepts one more character, stores the final word (one or two characters) in memory, and terminates reception. The EOT character is stored. If the controller is set for EOT-found disable, the receiver will not recognize the EOT character, and reception will continue until a word count overflow occurs. If no EOT character appears, reception will end with word count overflow.

At termination of reception, the receiver clears BUSY and sets DONE, requestion an interrupt.

Section 3.2 Transmitter Operation

To set up the transmitter to send data, the program must: supply an initial memory address for data words, supply a word count, and send a start command.

Setting SCT BUSY causes the transmitter to raise the Request To Send signal. When the local data set returns the Clear To Send signal, the transmitter requests a data channel access for the first word and begins sending the pairs of characters taken from the memory locations specified by the address counter. Since transmission is serial, the data channel has one bit time in which to respond to a request before sync is lost; if the channel is late, XMIT ERROR sets and a garbled pair of characters may be sent. Transmission continues to guarantee eventually reaching an EOT character at the end of the block.

If no errors occur during transmission, the word counter overflows as the last word is received from the channel; overflow clears BUSY and sets DONE, requesting an interrupt, even though the transmitter has one more word to send. The program then has two character times to supply a new initial address and word count and restart the transmitter before losing sync. If the transmitter is not restarted within this time, the Request To Send signal to the data set is dropped, and sync must be reestablished by transmitting a succession of sync characters before further data transmission can take place.

Section 3.3 Automatic Answer

The controller includes circuitry that allows the computer to answer incoming calls if the local data set is so configured. The program can sense data set control lines through use of bits 11-13 of the receiver status word. Status bit 11 (Carrier Detect) indicates that a carrier is being received from the remote station; bit 12 (Data Set Ready) indicates the local data set is connected and is capable of handling data.

Section 3.3 Automatic Answer (continued)

The program detects a ringing signal from a remote station by periodically examining bit 13, the Ring Indicator. The program answers a call by sending a Data Terminal Ready signal to the local data set; the program must also dismiss the call when completed.

Section 3.4 Internal Clock

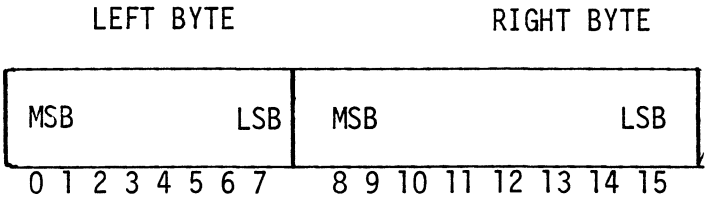
The controller board featured an internal clock to allow data exchanges to be timed when operating with an externally clocked data set or when a data link without a data set is used. The following speeds are available and are selectable via switches on the controller board (refer to section 5.0).

2400 baud
4800 baud
9600 baud
19,200 baud
38,400 baud

Section 4.0 Programming

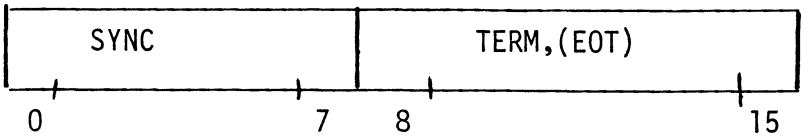
The instructions for the 340 Data Channel Sync Controller are in the standard I/O format. The primary device codes of receiver and transmitter are 40 and 41, and the mnemonics are SCR and SCT, respectively. The priority mask bit is 8 for both SCR and SCT.

Data channel transfers for both received and transmitted data contain two characters per word and are arranged as follows:



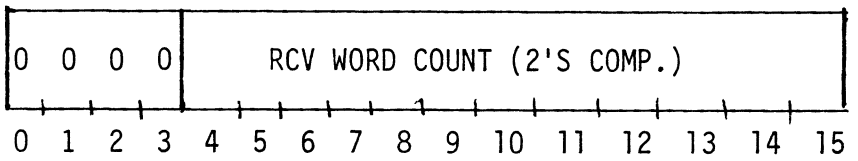
Section 4.1 Receiver Instructions

DOA <f> ac,SCR Defines the receiver sync and termination (EOT) characters according to the following format:

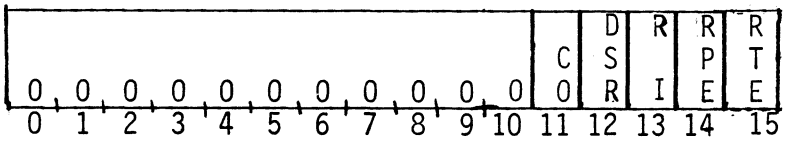


DOB <f> ac,SCR Loads the receiver address counter.

DOC <f> ac,SCR Loads the receiver word counter. Count loaded is in 2's complement form and a maximum block size of 4096 words (ac = 000000) may be loaded.



DIA <f> ac,SCR Reads the receiver status word as follows:



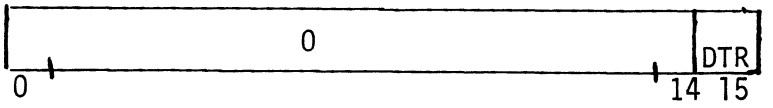
BIT	FUNCTION
11	Co-Carrier On-A carrier is being detected at the data set.
12	DSR-Data Set Ready - The data set is ready to accept data.
13	RI-Ring Indicator - The data set is receiving a ringing signal.
14	RPE-Receiver Parity Error - A character with incorrect parity has been received.
15	RTE-Receiver Timing Error - The computer data channel has failed to respond in time to a request for access by the receiver and incoming data has been lost.

Section 4.1 Receiver Instructions (Continued)

DIB<f> ac,SCR Reads the receiver address counter.

Section 4.2 Transmitter Instructions

DOA<f> ac,SCT Controls the data Terminal Ready line to the data set as follows.

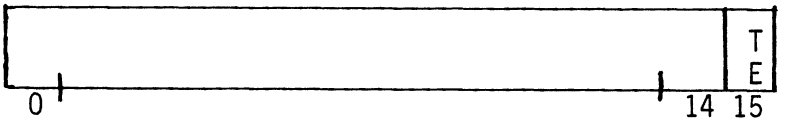


Bit 15=1 DTR sets
Bit 15=0 DTR resets

DOB<f> ac,SCT Loads the transmitter address counter.

DOC<f> ac,SCT Loads the transmitter word counter. (See receiver instruction for format.)

DIA<f> ac,SCT Reads the transmitter status word as follows:



BIT	FUNCTION
15	TE-Transmitter Error - The computer data channel has failed to respond in time to a request for access by the transmitter.

DIB<f> ac,SCT Reads the transmitter address counter.

Section 5.0 Configuring Operating Parameters

The following tables are used to set switches and jumpers on the controller board for selection of:

Device Code
Transmitter Character Size and Parity
Receiver Character Size and Parity
Internal Clock Speed
EOT Character Found Enable/Disable

Refer to Figure 1 in locating the switches and jumpers on the controller board.

For Switches,
O = open
C = closed
X = don't care

For Jumpers,
O = jumper out
C = jumper in
X = don't care

TABLE 1 DEVICE CODE SELECTION

Receiver Device Code	SW1-1	SW1-2	SW1-3	SW1-4
4X	C	C		
5X	C	O		
6X	O	C		
7X	O	O		
XO			C	C
X2			C	O
X4			O	C
X6			O	O

TABLE 2 INTERNAL CLOCK SPEED SELECTION

BAUD RATE	SW2-4	SW2-5	SW2-6	SW2-7	SW2-8
2.4K	O	O	O	O	C
4.8K	O	O	O	C	O
9.6K	O	O	C	O	O
19.2K	O	C	O	O	O
38.4K	C	O	O	O	O

TABLE 3 EOT CHARACTER FOUND ENABLE/DISABLE

SW2-1	SW2-2	
O	C	FOR EOT-FOUND ENABLE
C	O	FOR EOT-FOUND DISABLE

TABLE 4 RECEIVER CHARACTER SIZE AND PARITY SELECTION

JUMPER	CHARACTER FORMAT					
	6	7	8	6+P	7+P	8+P
J900	C	0	0	0	0	0
J901	0	C	C	C	C	C
J902	0	C	0	C	0	0
J903	C	0	0	0	0	0
J904	0	C	0	0	0	0
J905	C	0	0	C	0	0
J906	0	0	0	0	0	C
J907	0	0	C	0	C	0
J908	0	0	C	0	C	0
J909	0	0	0	0	0	0
J910	0	0	0	0	0	C
J911	0	0	C	0	0	0
J912	C	C	0	C	C	0
J91202	C	C	C	See table 5		

TABLE 5 RECEIVER PARITY SELECTION

JUMPER	PARITY			
	NO PARITY	ODD	EVEN	SUPPRESS EOT PARITY
J1201	X	O	C	X
J1202	C	O	O	X
J1203	X	C	O	X
J1204	X	X	X	C

TABLE 6 TRANSMITTER CHARACTER SIZE AND PARITY SELECTION

JUMPER	FORMAT					
	6	7	8	6+P	7+P	8+P
J601	O	C	C	O	C	C
J602	C	O	O	O	O	O
J603	O	O	O	C	O	O
J604	O	O	C	O	O	O
J605	O	O	O	O	C	C
J606	O	O	O	O	O	C
J607	O	O	C	O	C	O
J608	O	C	O	C	O	O
J609	C	O	O	O	O	O
J610	C	C	O	C	O	O
J611	O	O	O	O	O	C
J612	O	O	C	O	C	O
J613	O	C	O	O	O	O
J614	C	O	O	O	O	O
J615	O	O	C	O	O	O
J616	O	O	O	C	C	C
J617	O	O	O	O	O	C
J618	O	O	C	O	C	O
J619	O	C	O	C	O	O
J620	C	O	O	O	O	O
J621	C	C	O	C	O	O
J1207	X	X	X	O	C	C
J1208	X	X	X	O	O	C

TABLE 6 (continued)

JUMPER	FORMAT					
	6	7	8	6+P	7+P	8+P
J1211	X	X	X	O	C	C
J1212	X	X	X	O	O	C
J1213	X	X	X	C	O	O
J1214	X	X	X	C	C	O
J1215	X	X	X	C	O	O
J1216	X	X	X	C	C	O

TABLE 7 TRANSMITTER PARITY SELECTION

JUMPER	PARITY	
	ODD	EVEN
J1205	O	C
J1206	C	O
J1209	O	C
J1210	C	O

6.0 SPECIFICATIONS

POWER REQUIREMENTS

Power is supplied by the Nova +5 Volt and +15 power supplies.

+5 Volt Input Current= 3.5A Max
+15 Volt Input Current= 30MA Max

COMMUNICATIONS INTERFACE

Communications with the data set is in accordance with EIA standard RS-232C levels.

Transmit

Mark	-12 Volts Nominal
Space	+12 Volts Nominal

Receive

Mark	-3 to -25 Volts
Space	+3 to +25 Volts

ENVIRONMENTAL

Operating temperature	10° to 40°C
Operating humidity	10% to 90% NC
Non-operating temperature	-40°C to 55°C
Non-operating humidity	10% to 90%

7.0 INTERFACING

ALL SIGNALS ARE EIA STANDARD RS-232C

NOVA BACK PANEL	SIGNAL	50 PIN PADDLE BD	25 D (data set)
B50	Frame Ground	1	1
B53	Transmitted Data	46	2
B52	Received Data	45	3
B69	Request To Send	49	4
B54	Clear To Send	47	5
B40	Data Set Ready	41	6
B50	Signal Ground	1	7
B38	Carrier Detect	40	8
B34	Serial Clock Transmit	38	15
B51	Serial Clock Receiver	44	17
B49	Data Terminal Ready	43	20
B36	Ring Indicator	39	22
B67	Internal Clock	48	24

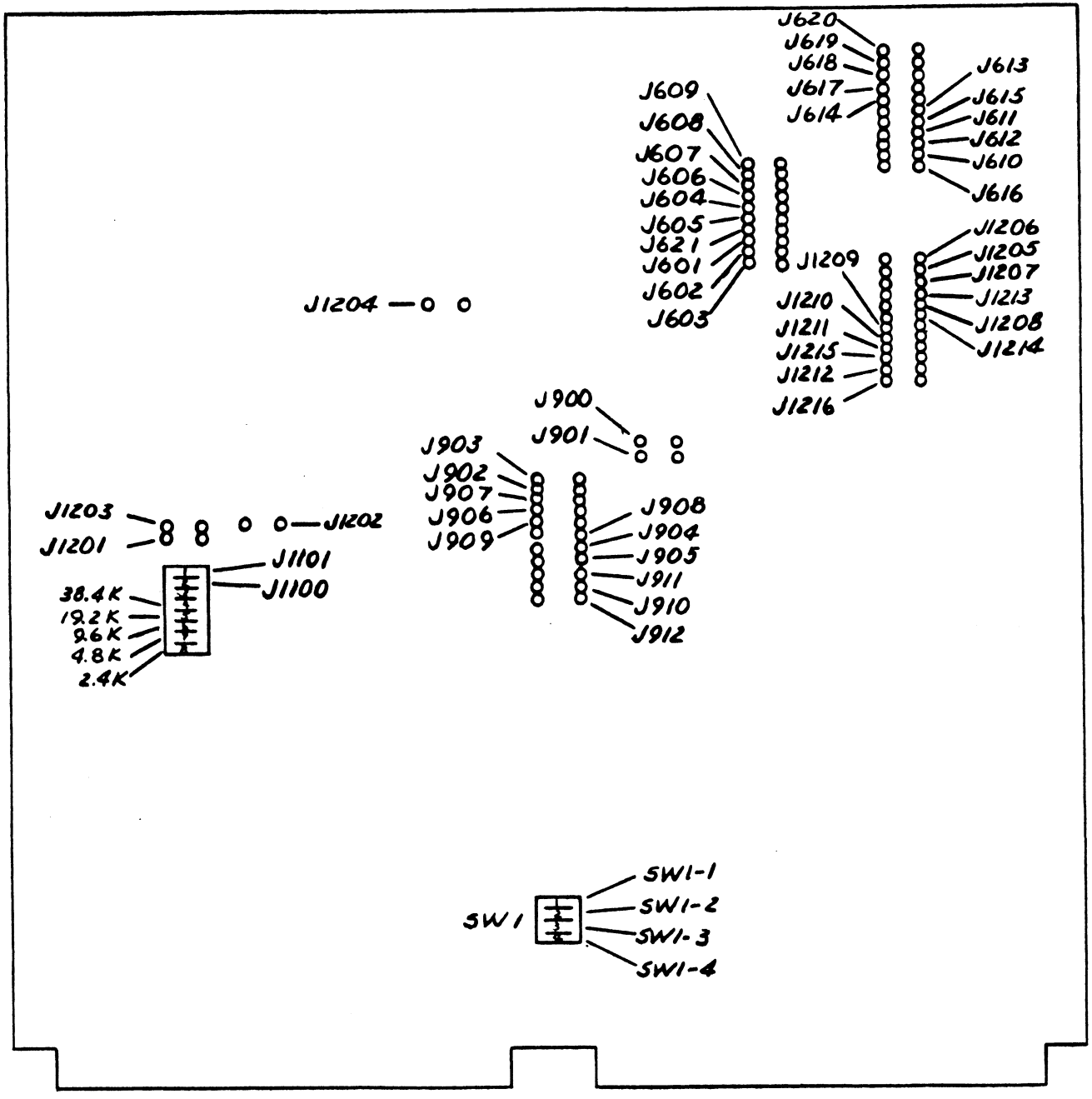


FIGURE 1
SWITCH AND JUMPER LOCATIONS

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.

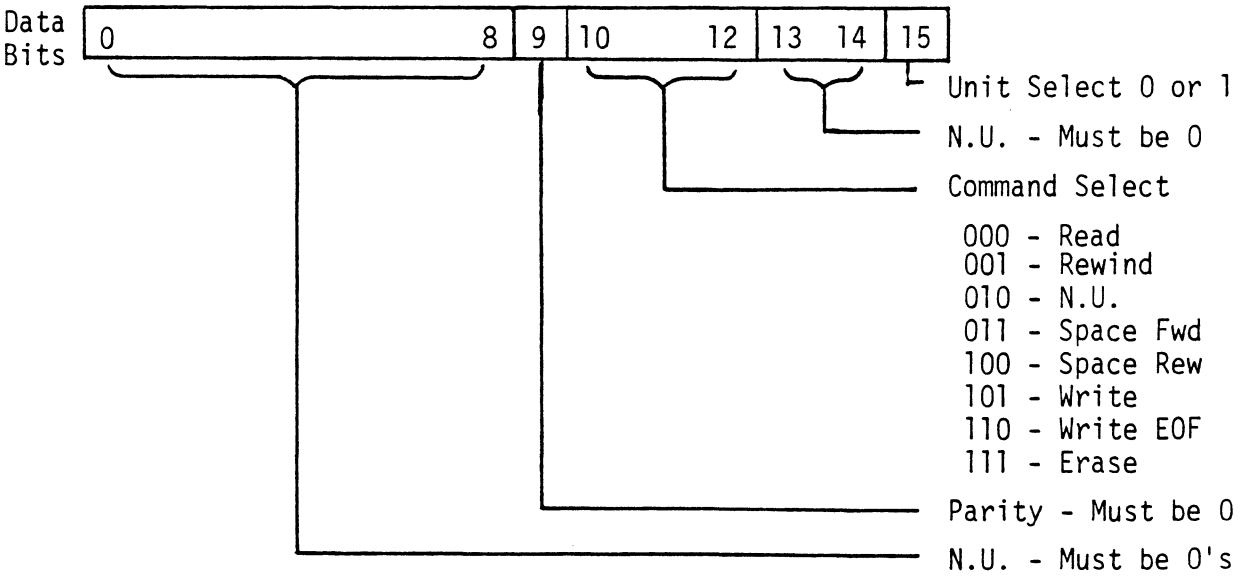
P (Pulse) - Not Used

Mask Bit 10

5.2 INSTRUCTIONS

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

Before execution, Accumulator AC should contain:



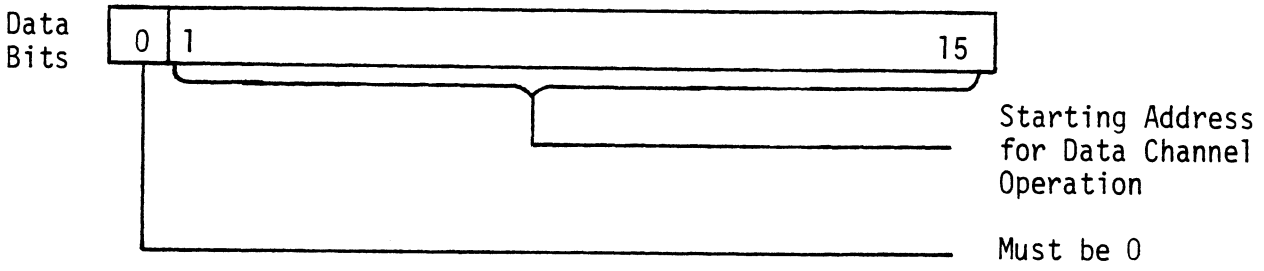
DOA(F) AC, MTA (continued)

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 -

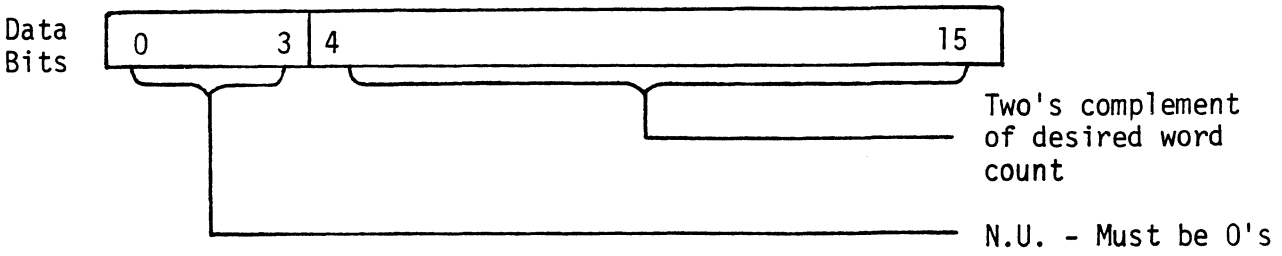
Before execution, Accumulator AC should contain:



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:

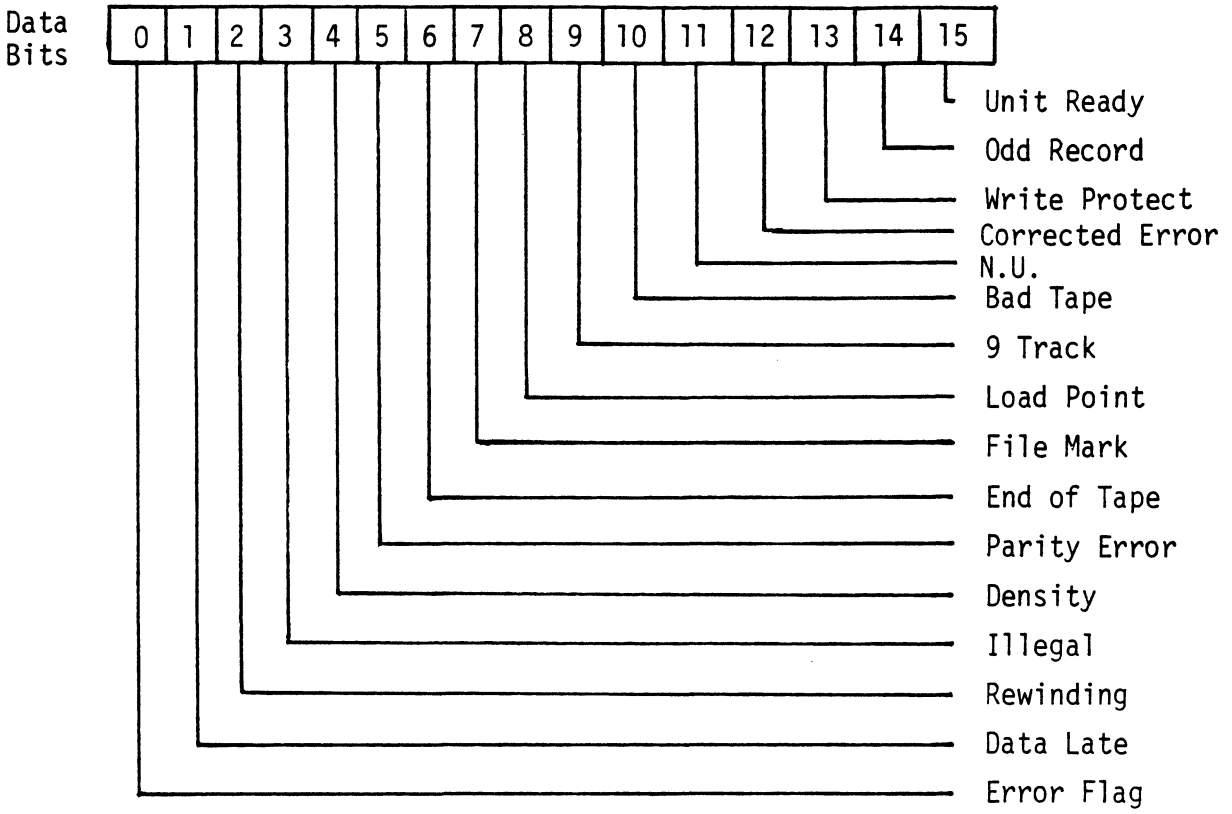


NOTE: Two's complement of the number of words to be transferred during the next Data Channel operation.

- 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.

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)

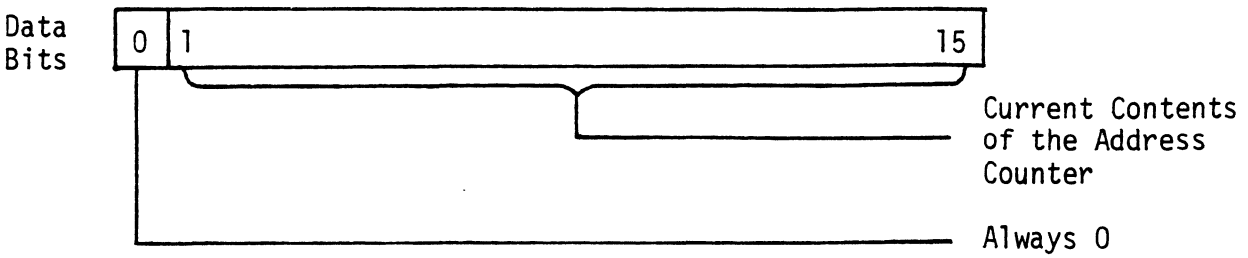
DATA BIT		SIGNIFICANCE
3	Illegal	- A start function was asserted under one of the following conditions: 1. A Write, Erase or Write FM issued with Write Protect on (Cartridge Lock in "Safe" position) 2. Space Reverse issued at Load Point 3. Space Forward, Write or Read issued with Unit at EOT 4. Unit 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)

7.2 DIAGNOSTICS (continued)

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. OX

  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)
```

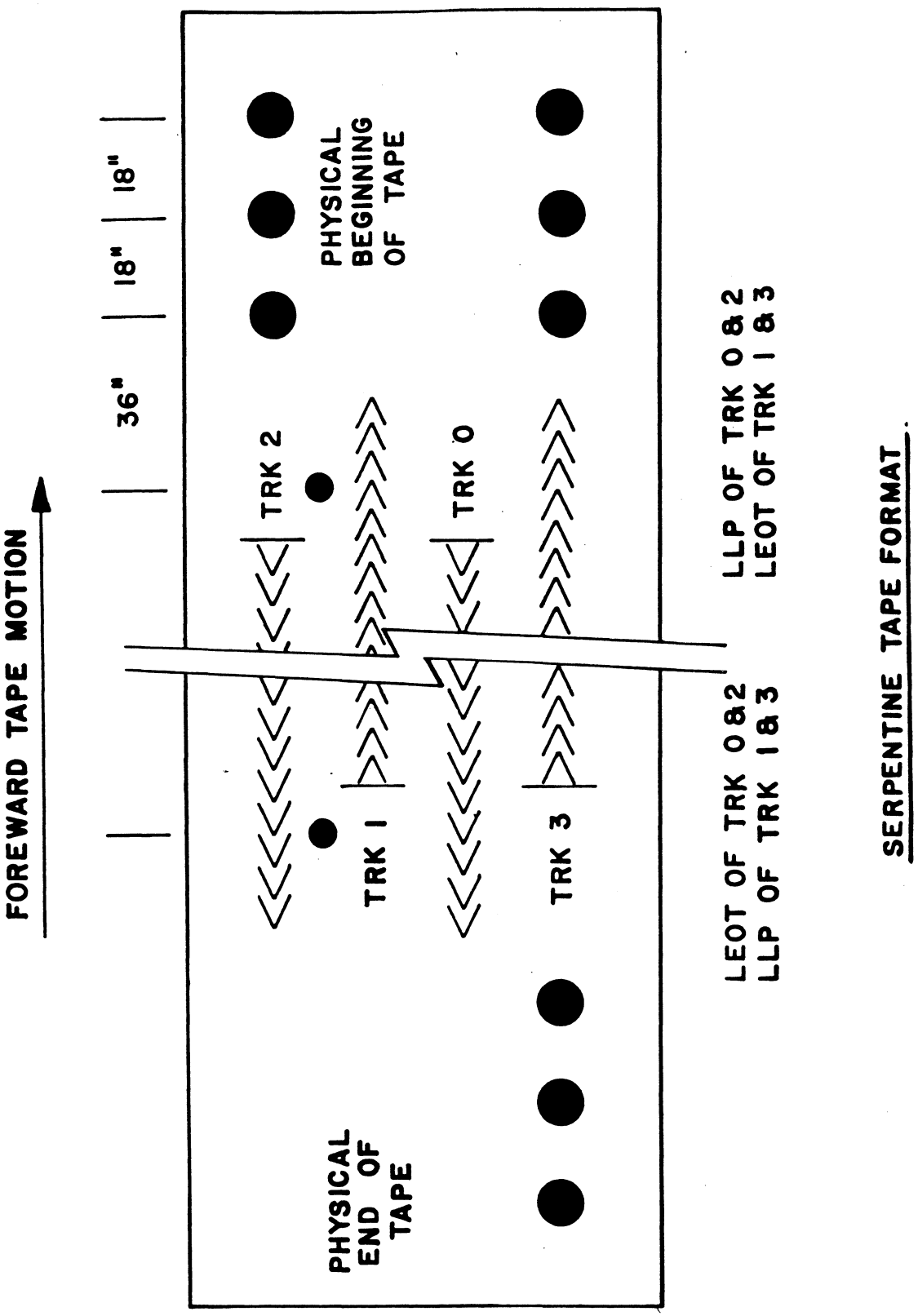
This test will print out any detected error whether it recovered or not. If it recovered the error, it will print how many passes it took to recover it. If it did not recover the error will print "Hard Error" after the error report.

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).

```
 0001 MT110    MACRO REV 06.70         10:24:23 08/10/82
01              ;
02              ;
03              ;
04              ;*****************************************************************
05              ;
06              ;
07              ;
08              ; DESCRIPTION: MT110 1/4" MAGNETIC TAPE DIAGNOSTIC
09              ;
10              ;
11              ; CUSTOM SYSTEMS INC,  1982
12              ;*****************************************************************
                        .TITL   MT110D
14       000001         .DUSR   X=1
15       000001         .NOMAC  X
16              ;1.      PROGRAM NAME: MT110D.SR
17              ;2.     REVISION HISTORY:
18              ;
19              ;               REV.      DATE
20  00422 000000        .REV    0,0     ;04/27/82
21              ;
22              ;3.     MACHINE REQUIREMENTS
23              ;       3.1     NOVA OR ECLIPSE FAMILY CPU'S
24              ;       3.2     MINIMUM OF 16K MEMORY
25              ;       3.3     6450 KENNEDY 1/4" CARTRIDGE TAPE DRIVE
26              ;       3.4     MT110 CONTROLLER BOARD
27              ;       3.5     TELETYPE OR CRT AND CONTROLLER
28              ;4.     TEST REQUIREMENTS
29              ;       N/A
30              ;5.     SUMMARY
31              ;       THIS PROGRAM IS A HARDWARE DIAGNOSTIC FOR THE MODEL 110
32              ;       CONTROLLER.  THE DEVICE CODE CAN BE SELECTED AS
33              ;       22 OR 62.  ONLY ONE(1) READY, WRITE ENABLED DRIVE CAN BE
34              ;       ON LINE AT A TIME.
35              ;
36              ;6.     RESTRICTIONS
37              ;
38              ;       ONLY ONE(1) DRIVE CAN BE ONLINE AT ANY TIME.  THE DEVICE
39              ;       CODE MUST BE SELECTED AS 22 OR 62.  ALL RESPONSES TO PRO-
40              ;       GRAM REQUESTS MUST BE ANSWERED PROPERLY TO CONTINUE THE
41              ;       SEQUENTIAL TESTING OF THE TAPE DRIVE.
```

```
!0002 MT110
01              ;7.     PROGRAM DESCRIPTION/THEORY OF OPERATION
02              ;       7.1     INITIALIZATION
03              ;               7.1.1   I/O MODULE INITIALIZED
04              ;               7.1.2   TEST SELB LINE SET, IF LINE SET IR-
05              ;                       RECOVERABLE ERROR.  PROGRAM HALTS AT
06              ;                       BHALT.
07              ;                               1.      SELECT UNIT NUMBER
08              ;                               2.      DEVICE CODE CHANGE
09              ;                               3.      SET SOFT SWITCH REGISTER
10              ;       7.2     PRELIMINARY TESTS
11              ;               7.2.1   TEST A1 - TEST SYSTEM SELD LINE.
12              ;               7.2.2   TESTS A2 AND A3 - TEST CONTROLLER BUSY
13              ;                       AND DONE STATUS.
14              ;               7.2.3   TESTS A4 THRU A8 - TEST FOR UNIT SELECT
15              ;                       BY LOADING AND TESTING THE MEMORY
16              ;                       ADDRESS REGISTER.
17              ;               7.2.4   TESTS A9 THRU A14 - TEST FOR UNIT READY
18              ;                       AND THE SETTING AND RESETTING OF BUSY
19              ;                       AND DONE BY THE START COMMAND.
20              ;       7.3     FIRST TAPE MOTION
21              ;               7.3.1   TESTS A15 AND A16 - TEST REWIND AND
22              ;                       ERASE OPERATION AND STATUS.
23              ;       7.4     FIRST DATA TRANSFER
24              ;               7.4.4   TESTS A20 AND A21 - TEST FOR TOTAL DATA
25              ;                       DATA WRITE WITH INTERRUPT.
26              ;               7.4.5   TESTS A22 AND A23 - TEST WRITE EVEN AND
27              ;                       ODD PARITY.
28              ;       7.5     STATUS BIT TESTS
29              ;               7.5.1   TEST A24 AND A25 - TEST FOR ILLEGAL
30              ;                       COMMAND STATUS BIT SETTING.
31              ;               7.5.2   TEST A26 - TEST FOR EOF STATUS BIT
32              ;                       SETTING.
33              ;               7.5.4   TESTS A28 THRU A 32 - TEST STATUS BITS
34              ;                       AND MEMORY ADDRESS REGISTER DURING BACK
35              ;                       AND FORWARD SPACING.
36              ;       7.6     DATA TRANSFER TESTS
37              ;               7.6.1   TESTS A33 AND A34 - TEST WRITE AND READ
38              ;                       IN ODD AND EVEN PARITY.
39              ;               7.6.2   TESTS A35 AND A36 - TEST WRITE AND READ
40              ;                       WITH DIFFERENT WORD COUNTS.
41              ;               7.6.4   TESTS A39 THRU A41 - TEST EOF WRITE AND
42              ;                       READ.
43              ;               7.6.6   TESTS A50 THRU A53 - TEST FOR SPACING
44              ;                       ERRORS BY GENERATING NOISE WITH I/O
45              ;                       COMMANDS.
46              ;               7.6.7   TEST A54 - TEST LONGITUDINAL PARITY
47              ;                       ERRORS BY WRITTING EVEN PARITY, ZERO
48              ;                       DATA PATTERNS.
49              ;       7.7     WRITE LOCK TEST
50              ;               THIS TEST DETERMINES IF THE CARTRIDGE SET TO SAFE
51              ;               WILL DISABLE THE WRITE.  THIS TEST IS ONLY
52              ;               PERFORMED DURING THE FIRST PASS AND CAN BE DE-
53              ;               LETED BY SETTING SOFT SWITCH REGISTER BIT 15.
54              ;       7.8     END OF TAPE TEST
55              ;               THIS TEST WRITES 4K BLOCKS FROM BOT TO EOT.  DUR-
56              ;               ING THE TAPE WRITE ALL ERROR STATUS CONDITIONS
57              ;               ARE MONITORED.  WHEN THE EOT SENSOR IS DETECTED
58              ;               THE WRITE OPERATION IS TERMINATED AND THE TAPE IS
59              ;               COMMANDED TO REWIND.  IF THE EOT SENSOR IS NOT DE-
60              ;               TECTED THE WRITE WILL CONTINUE UNTIL THE TAPE
```

```
01              ;               COMES OFF THE SUPPLY REEL.  THIS TEST CAN BE DE-
02              ;               LETED BY SETTING SOFT SWITCH REGISTER BIT 14.
03              ;8.     SOFT SWITCH REGISTER SETTINGS
04                      S?WPD 8
05              ;       8.3     SWITCH OPTIONS
06              ;
07              ;               DIFFERENT SWITCH BITS AND THEIR INTERPRETATION
08              ;                AT LOCATION "SWREG" ARE AS FOLLOWS:
09              ;
10              ;               BIT     OCTAL   BINARY  INTERPRETATION
11              ;                       VALUE   VALUE
12              ;               14(E)   00002   0       ENABLE WRITE TO EOT TEST
13              ;                               1       INHIBIT WRITE TO EOT TEST
14              ;
15              ;               15(F)   00001   0       ENABLE WRITE LOCK TEST
16              ;                               1       INHIBIT WRITE LOCK TEST
17              ;
18              ;               NOTE:   SWITCH BITS 14 AND 15 CAN ONLY BE
19              ;                       ENABLED DURING THE FIRST PASS OF THE
20              ;                       DIAGNOSTIC.  IF THE TESTS ARE TO BE PER-
21              ;                       FERMED AFTER THE FIRST PASS, THEY CAN BE
22              ;                       DIRECTLY ENTERED.
23              ;9.     OPERATING PROCEDURES
24              ;       9.1     PROGRAM LOAD
25              ;               LOAD THE PROGRAM BY USING THE BINARY LOADER.
26              ;       9.2     STARTING ADDRESSES
27              ;               201     DIRECT ENTRY TO OCTAL DEBUGGER(ODT)
28              ;               500     START DIAGNOSTIC
29              ;               501     DIRECT ENTRY TO WRITE LOCK TEST
30              ;               502     DIRECT ENTRY TO WRITE TO EOT TEST
31              ;       9.3     PROGRAM OPERATION
32              ;               9.3.1   THE FOLLOWING HEADER IS PRINTED.
33              ;       "C.S.I.  MT110 DIAG. (KENNEDY 1/4") REV.  XX"
34              ;       "MOUNT A WRITE ENABLED SCRATCH TAPE, HIT ANY KEY TO CONTINUE"
35              ;               9.3.2   A UNIT REQUEST IS MADE AS FOLLOWS:
36              ;       "DRIVE UNIT #:"
37              ;                       RESPOND TO THE REQUEST BY TYPING A
38              ;                       UNIT NUMBER 0 OR 1 FOLLOWED BY A
39              ;                       CARRIAGE RETURN.
40              ;               9.3.5   A REQUEST FOR A DEVICE CODE IS MADE
41              ;                       AS FOLLOWS:
42              ;       "MTA DEVICE CODE:"
43              ;                       RESPOND TO THE REQUEST BY TYPING EITHER
44              ;                       22 OR 62.  ANY OTHER INPUT WILL CAUSE AN
45              ;                       ERROR.
46              ;               9.3.6   A REQUEST TO MODIFY THE SOFT SWITCH
47              ;                       REGISTER IS MADE AS FOLLOWS:
48              ;       "SET SWITCH REGISTER TO DESIRED VALUE, TYPE CR TO CONTINUE."
49              ;                       THE PROGRAM WILL BE LOCKED IN THE MODIF-
50              ;                       ICATION MODE.  SET ALL BITS TO THEIR COR-
51              ;                       RECT STATE BY HITTING THE APPROPRIATE
52              ;                       KEY.  TO TERMINATE THE SWITCH MODIFICATION
53              ;                       MODE TYPE A CARRIAGE RETURN.
54              ;               9.3.7   IF A SYSTEM REAL TIME CLOCK IS NOT PRE-
55              ;                       SENT, THE FOLLOWING REQUEST WILL BE
56              ;                       PRINTED.
57              ;       "TTO BAUD RATE = ? "
58              ;                       RESPOND TO THE REQUEST BY TYPING THE
59              ;                       CORRECT CONSOLE DEVICE BAUD RATE FOR THE
60              ;                       I/O TIMING CALIBRATION.  IF THE RESPONSE
```

```
01              ;                       IS 110, THE FOLLOWING REQUEST MESSAGE
02              ;                       WILL BE PRINTED.
03              ;       "(ONLY DASHER AT 110 BAUD = 10 BITS/CHAR"
04              ;       "OTHERWISE = 11 BITS/CHAR.)"
05              ;       "# BITS/CHAR = ? "
06              ;                       RESPOND TO THE REQUEST BY TYPING 10 OR
07              ;                       11.
08              ;               9.3.8   THE DIAGNOSTIC TESTING WILL TEST THE
09              ;                       CONTROLLER AND SELECTED DRIVE.  ALL DE-
10              ;                       TECTED FAULTS WILL BE PRINTED AS SPEC-
11              ;                       IFIED BY SECTION 10 AND PROGRAM CON-
12              ;                       TROL IS DETERMINEDBY THE SETTING OF THE
13              ;                       SOFT SWITCH REGISTER.  EACH SUCCESSIVE
14              ;                       TEST ASSUMES THAT ALL PREVIOUS TESTS
15              ;                       HAVE PASSED CORRECTLY.  BYPASSING A
16              ;                       FAILING TEST MAY LEAD TO CONFUSING
17              ;                       RESULTS ON SUCCEDING TESTS.
18              ;
19              ;               9.3.9   WRITE LOCK TEST
20              ;                       THE WRITE LOCK TEST IS ENTERED DURING
21              ;                       THE FIRST PASS OF THE DIAGNOSTIC OR BY
22              ;                       DIRECT ENTRY.  THE SELECTED UNIT IS
23              ;                       TESTED BY ATTEMPTING TO WRITE WHEN THE
24              ;                       CARTRIDGE IS SET TO SAFE.  THE FOLLOW-
25              ;                       ING REQUEST IS PRINTED AT THE START OF
26              ;                       THE TEST.
27              ;       "MOVE CARTRIDGE SWITCH TO THE SAFE POSITION.  DONT STOP THE PROGRAM."
28              ;                       RESPOND TO THE REQUEST AS FOLLOWS:
29              ;                               MOVE CARTRIDGE SWITCH TO SAFE
30              ;                       WHEN THE DRIVE COMES READY, AN ATTEMPT
31              ;                       IS MADE TO WRITE A RECORD.  IF THE WRITE
32              ;                       IS INHIBITED, THE FOLLOWING REQUEST IS
33              ;                       PRINTED.
34              ;       "MOVE CARTRIDGE SWITCH OFF THE SAFE POSITION."
35              ;                       MOVE SWITCH OFF THE SAFE POSITION
36              ;               9.3.10  WRITE TO EOT SENSOR
37              ;                       THE TEST IS PERFORMED DURING THE FIRST
38              ;                       PASS OF THE DIAGNOSTIC OR BY DIRECT
39              ;                       ENTRY.  THE TEST WRITES 4K DATA BLOCKS
40              ;                       FROM BOT TO THE EOT SENSOR.  ALL STATUS
41              ;                       BITS ARE EXAMINED DURING EACH WRITE AND
42              ;                       IF ANY FAULTS ARE DETECTED AN APPROPRI-
43              ;                       ATE STATUS ERROR MESSAGE IS PRINTED.  IF
44              ;                       THE EOT SENSOR IS NOT DETECTED THE WRITE
45              ;                       WILL CONTINUE UNTIL THE TAPE COMES OFF
46              ;                       THE SUPPLY REEL INDICATING THE ERROR.  IF
47              ;                       THE EOT SENSOR IS DETECTED, THE FOLLOW-
48              ;                       ING MESSAGE IS PRINTED.
49              ;       "END OF TAPE"
50              ;       "CYCLE"
51              ;                       AFTER THE MESSAGE IS PRINTED, THE DRIVE
52              ;                       IS COMMANDED TO REWIND.  WHEN THE REWIND
53              ;                       OPERATION IS COMPLETE THE FOLLOWING MES-
54              ;                       SAGE IS PRINTED.
55              ;       "PASS 1"
56              ;10.    PROGRAM ERROR DESCRIPTION
57              ;       10.1    PRELIMINARY TEST ERRORS
58              ;               THE FOLLOWING IS A LIST OF PRELIMINARY CON-
59              ;               TROLLER AND DRIVE ERROR MESSAGES.
60              ;               10.1.1  BUSY AND DONE ERRORS
```

```
01              ;       "SELD LINE NOT RESET BY IORST, PC = XXXXX"
02              ;       "BUSY FLIP-FLOP NOT RESET ERROR, PC = XXXXX"
03              ;       "BUSY FLIP-FLOP NOT RESET BY IORST, PC = XXXXX"
04              ;       "BUSY FLIP-FLOP NOT SET ERROR, PC = XXXXX"
05              ;       "DONE FLIP-FLOP NOT RESET ERROR, PC = XXXXX"
06              ;       "DONE FLIP-FLOP NOT SET ERROR, PC = XXXXX"
07              ;               10.1.2  CONTROLLER DATA TRANSFER ERRORS
08              ;       "SEND CLOCK BIT ON TOO LONG ERROR, PC = XXXXX"
09              ;       "FIRST CHARACTER TIME OUT ERROR, PC = XXXXX"
10              ;       "DATA TRANSFER TIME OUT ERROR, PC = XXXXX"
11              ;       "NO INTERRUPT ERROR, PC = XXXXX"
12              ;       "ILLEGAL INTERRUPT WITH MASK BIT SET, MASK = XX,/
13              ;       PC = XXXXX"
14              ;       "MTU SELECT ERROR, DIB COMMAND = XXXXXX, PC = XXXXX"
15              ;       "MA REGISTER NOT RESET BY IORST"
16              ;       "GOOD WORD = XXXXXX, BAD WORD = XXXXXX, PC = XXXXX"
17              ;       "MA REGISTER SETTING ERROR"
18              ;       "GOOD WORD = XXXXXX, BAD WORD = XXXXXX, PC = XXXXX"
19              ;       "INTA DEVICE CODE ERROR"
20              ;       "DEVICE CODE = XX, UNIT DEVICE CODE = XX, PC = XXXXX"
21              ;       10.2    SYSTEM ERRORS
22              ;               THE FOLLOWING ERRORS OCCURE DURING COMBINED CON-
23              ;               TROLLER AND DRIVE OPERATIONS.
24              ;               10.2.1  DATA TRANSFER AND MA REGISTER ERRORS
25              ;       "MA REGISTER COUNTING ERROR"
26              ;       "GOOD VALUE = XXXXXX, BAD VALUE = XXXXXX, PC = XXXXX"
27              ;       "DATA COMPARE ERROR"
28              ;       "GOOD WORD = XXXXXX, BAD WORD = XXXXXX, /
29              ;       MEMORY ADDRESS = XXXXXX, PC = XXXXX"
30              ;               10.2.2  STATUS ERRORS
31              ;       "EXPECTED STATUS = XXXXXX, ACTUAL STATUS = XXXXXX, /
32              ;       PC = XXXXX"
33              ;       10.3    STATUS WORD
34              ;               BIT     DESCRIPTION
35              ;               0       ANY ERROR, SET BY BITS 1,3,5,6,7,8,10,14
36              ;               1       DATA LATE
37              ;               2       REWINDING
38              ;               3       ILLEGAL COMMAND
39              ;               4       HIGH DENSITY
40              ;               5       PARITY ERROR
41              ;               6       EOT MARK SENSED
42              ;               7       EOF MARK SENSED
43              ;               8       BOT MARK SENSED
44              ;               9       9 TRACK TAPE
45              ;               10      BAD TAPE
46              ;               11      SEND CLOCK
47              ;               12      FIRST CHARCTER
48              ;               13      WRITE LOCKOUT
49              ;               14      CRC ERROR
50              ;               15      UNIT READY
51                      ;       O7DTD   11
52              ;12.    SPECIAL NOTES
53              ;       12.1    MEDIA SELECTION
54              ;               IT IS IMPORTANT TO SELECT A KNOWN GOOD TAPE WHEN
55              ;               PERFORMING THE DIAGNOSTIC.  ANY ERRORS CAUSED BY
56              ;               THE MEDIA WILL BE CONSIDERED A CONTROLLER AND/OR
57              ;               DRIVE FAULT.
58              ;       12.2    SCOPE LOOPS
59              ;               WHEN A SCOPE LOOP IS BEING IMPLEMENTED TO LOCATE
60              ;               A FAILING MODULE AND FORWARD TAPE MOTION IS
```

```
01              ;               USED, THE TAPE WILL COME OFF THE SUPPLY REEL IF
02              ;               THE LOOP IS ALLOWED TO CONTINUE.  WHEN THE TAPE
03              ;               APPROACHES THE EOT SENSOR, ENTER THE ODT PROGRAM
04              ;               BY TYPEING A CONTROL "O" CHARACTER, MANUALLY RE-
05              ;               WIND THE DRIVE AND TYPE A "P" CHARACTER TO CON-
06              ;               TINUE.
07              ;13.    RUN TIME
08              ;       THE PROGRAM RUN TIME IS DEPENDENT ON THE
09              ;       LENGTH OF THE TAPE.
10              ;       IT IS RECOMMENDED THAT A
11              ;       300 FOOT CARTRIDGE BE USED TO FACILITATE A FAST
12              ;       WRITE TO THE EOT SENSOR TEST.
```

```
 0007 MT110
01
02       000000          .NOMAC  0
03       000000          .LOC 0
04 00000 000000          0
05 00001 000010          INTR
06 00002 000200          DTO?SB
07 00003 002002          JMP     @.-1
08 00004 000000          0
09 00005 000000          0
10
11               ; IRRECOVERABLE ERROR HALT, SELB LINE ALWAYS SET
12
13 00006 063077 BHALT:   HALT                    ; HALT FOR HARD SELB LINE SET
14 00007 000200          JMP     DTO?SB          ; START PROGRAM AGAIN ON CONTINUE
15 00010 010000 INTR:    ISZ     0
16 00011 002000          JMP     @0
17 00012 002226          JMP     @IODT?          ;GO TO ODT
18
19       000045          .LOC    45              ;DO NOT INSERT,LOC 45
20                                                ;CONTAINS EGGS POINTER
21 00045 006757 NEST:    EGGS
22
23       000050          .LOC    50
24
25 00050 000010          .BLK    10              ; RESERVED FOR SYSTEM DEBUG
26
27                       ;CONSTANTS
28 00060 000503 FIRST:   DIAG
29 00061 007016 CEND:    LAST
30 00062 075402 CMTA:    DIB 3,2
31 00063 007016 OBUFF:   LAST
32 00064 007216 IBUFF:   LAST+200
33 00065 000031 .25MS:   25.
34 00066 000062 .50MS:   50.
35 00067 000310 .200MS:  200.
36 00070 000620 .400MS:  400.
37 00071 010421 C10421:  10421
38 00072 021042 C21042:  21042
39 00073 042104 C42104:  42104
40 00074 004210 C421T:   4210
41 00075 160077 C160077:         160077
42 00076 003265 PCDCM:   CDCM
43 00077 177700 C177700:         177700
44 00100 110000 C110000:         110000
45 00101 125252 C125252:         125252
46 00102 052525 C52525:  52525
47 00103 000144 DL100:   100.
48 00104 001064 CRCNT:   500.+100
49
50 00105 000001 C1:      1
51 00106 000002 C2:      2
52 00107 000003 C3:      3
53 00110 000004 C4:      4
54 00111 000005 C5:      5
55 00112 000006 C6:      6
56 00113 000007 C7:      7
57 00114 000010 C10:     10
58 00115 000011 C11:     11
59 00116 000012 C12:     12
60 00117 000013 C13:     13
```

```
 0008 ;T110
01 00120 000014 C14:    14
02 00121 000015 C15:    15
03 00122 000016 C16:    16
04 00123 000017 C17:    17
05 00124 000020 C20:    20
06 00125 000024 C20.:   20.
07 00126 000021 C21:    21
08 00127 000022 C22:    22
09 00130 000030 C30:    30
10 00131 000040 C40:    40
11 00132 000060 C60:    60
12 00133 000062 C62:    62
13 00134 000070 C70:    70
14 00135 000077 C77:    77
15 00136 000100 C100:   100
16       000103 C100.=DL100
17 00137 000101 C101:   101
18 00140 000102 C102:   102
19 00141 000103 C103:   103
20 00142 000104 C104:   104
21 00143 000144 C144:   144
22 00144 000150 C150:   150
23 00145 000200 C200:   200
24 00146 000201 C201:   201
25 00147 001000 C1000:  1000
26 00150 002000 C2000:  2000
27 00151 020000 C20K:   20000
28 00152 100401 C1004:  100401
29 00153 004000 C4K:    4000
30 00154 007700 C7700:  7700
31 00155 037477 C37477: 37477
32 00156 040000 C40000: 40000
33       000150 C2MS=C2000
34
35 00157 005670 D3000:  3000.
36 00160 000764 D500:   500.
37
38 00161 177775 M3:     -3
39 00162 177774 M4:     -4
40 00163 177770 M8:     -8.
41 00164 177700 M100:   -100
42 00165 177160 M400:   -400.
43 00166 000116 ASCN:   116
44 00167 000131 ASCY:   131
45 00170 000000 IIDX0:  0               ;%
46 00171 000000 IIDX1:  0               ;%
47 00172 000000 IIDX2:  0               ;%
48 00173 000000 IIDX3:  0               ;%
49 00174 000000 IIDX4:  0               ;%
50
51              ; POINTERS AND CONSTANTS
52
53       000200         .LOC 200
54
55       000000         .DUSR   COM?P=0
56
57 00200 002202 DTO?SB: JMP @BGNADR             ;PROGRAM START
58 00201 002226         JMP @IODT?              ;DIRECT ENTER TO ODT
59 00202 000500 BGNADR: BEGIN                   ; PROGRAM STARTING ADDRESS
60 00203 000000 PAS?S:  0                       ; PASS COUNTER
```

```
 0009 MT110
01 00204 000000 ODO?K:  0                       ; ODT BREAKPOINT LOCATION
02
03              ; PAGE ZERO POINTERS
04
05 00205 006764 ISWR?EG: SWREG                  ; SWITCH REGISTER POINTER
06 00206 005750 IINP?:  INP?J                   ; SWITCH PACK POINTER
07 00207 005226 IMES?:  MES?S                   ; MESSAGE PRINT POINTER
08 00210 005275 ICRL?F: CRL?F                   ; CR/LF PRINT ROUTINE POINTER
09 00211 005424 ITYP?E: TYP?E                   ; CHARACTER PRINT ROUTINE POINTER
10 00212 005330 IPDE?C: PDE?C                   ; DECIMAL PRINT ROUTINE POINTER
11 00213 005320 IPDC?S: PDC?S                   ; DECIMAL AND 1 CHAR. PRINT
12 00214 005312 IPOC?T: POC?T                   ; OCTAL PRINT ROUTINE POINTER
13 00215 005306 IZOC?T: ZOC?T                   ; ZERO SURPRESSED OCTAL PRINT
14 00216 005570 ITI?O:  TIN?O                   ; OCTAL INPUT ROUTINE POINTER
15 00217 005574 ITI?D:  TIN?D                   ; DECIMAL INPUT ROUTINE POINTER
16 00220 005421 ITPS?P: TPS?P                   ; TYPE SPACE ROUTINE POINTER
17 00221 000000 IOM?OD: 0                       ; I/O MODULE POINTER
18 00222 005670 ITI?A:  TIN?A                   ; INPUT CHARATER ROUTINE POINTER
19 00223 006420 ICAL?B: CAL?B                   ; CALIBRATE TIME BASE POINTER
20 00224 006624 IDEL?Y: DEL?Y                   ; TIME DELAY POINTER
21 00225 006642 ITIM?R: TIM?R                   ; TIMER TEST POINTER
22 00226 006212 IODT?:  ODT?J                   ; OCTAL DEBUGGER POINTER
23 00227 006721 IMLT?:  MUL?T                   ; MULTIPLY ROUTINE POINTER
24 00230 006741 IDVI?:  DIV?I                   ; DIVIDE ROUTINE POINTER
25 00231 005533 IPC7?:  PC7?                    ; TAB COUNTER POINTER
26
27                      ;VARIABLES
28
29 00232 000000 X70:    0
30 00233 000010 C1X:    10                      ;UNIT # +10
31 00234 000000 CX:     0                       ;UNIT #
32 00235 000050 C5X:    50                      ;UNIT # +50
33 00236 000000 CC5X:   0                       ;UNIT # +50
34 00237 000070 C7X:    70                      ; UNIT # + 70
35 00240 000000 CNTR:   0                       ;COUNTER
36 00241 000000 WDCNT:  0                       ;TEMP STORAGE
37 00242 000000 INST:   0                       ;  "     "
38 00243 000022 DEVICE: .MTA                    ;DEVICE CODE.  22 OR 62
39 00244 000000 RECTR:  0
40 00245 000000 WCTR:   0
41
42                      ;PATTERN TABLE, ODD PARITY WRITES
43
44 00246 000246 POINT:  .
45 00247 000000         0
46 00250 177400         177400
47 00251 000377         377
48 00252 125252         125252
49 00253 052525         52525
50 00254 000253 LST:    .-1
51
52                      ;PATTERN TABLE, EVEN PARITY WRITES
53
54 00255 000255 PONTR:  .
55 00256 125252         125252
56 00257 052525         052525
57 00260 000401         000401
58 00261 020040         020040
59 00262 000261 LST1:   .-1
60
```

```
 0010 MT110
01                          ;INDIRECT ADDRESSES
02 00263 003165 ICKST:  CSTAT
03 00264 002431 INTIL:  INITIAL
04 00265 003207 ISTAT:  XSTAT
05 00266 003144 IERAS:  XERAS
06 00267 003072 IWRT:   XWRT
07 00270 002763 IGEN:   XGEN
08 00271 002722 IWEOF:  XWEOF
09 00272 002615 IBSP:   XBSP
10 00273 003067 IWNS:   XWNS
11 00274 004557 ABTBF:  BTBF
12 00275 004617 SETP1:  ENTER-3
13 00276 004622 SETP:   ENTER
14 00277 004640 LOOP.:  CYCLE
15 00300 004722 ER:     ERR
16 00301 003015 IRD:    XRD
17 00302 002577 ICLR:   XCLR
18 00303 002547 ICHK:   XCHK
19 00304 002743 ILD:    XLD
20 00305 002620 ISPC:   XSPC
21 00306 003002 ISEL:   XSEL
22 00307 003274 PINWL:  INWLT
23 00310 003233 PETTA:  ETTA
24 00311 003266 IWLT:   WLT
25 00312 003224 IETT:   ETT
26 00313 005202 ITEST:  TSTIN
27 00314 004714 ISPE1:  SPER1
28 00315 004717 ISPE2:  SPER2
29                          ;DEFINITIONS
30       000022 .MTA=22
31       006265 STATUS=JSR @ISTAT
32       006266 ERASE=JSR @IERAS
33       006267 WRITE=JSR @IWRT
34       006270 GEN=JSR @IGEN
35       006271 WEOF=JSR @IWEOF
36       006272 BSPACE=JSR @IBSP
37       006273 WRTNS=JSR @IWNS
38       006224 WAIT=JSR @IDEL?Y
39       006305 SPACE=JSR @ISPC
40       006300 EHALT=JSR @ER
41       006314 HALT1=JSR @ISPE1
42       006315 HALT2=JSR @ISPE2
43       006301 READ=JSR @IRD
44       006302 CLEAR=JSR @ICLR
45       006303 CHECK=JSR @ICHK
46       006304 LOAD=JSR @ILD
47       006306 SELECT=JSR @ISEL
48 00316 002656 .RWNS
49       006316 RWNS=JSR @.-1
50 00317 002661 .RW
51       006317 REWIND=JSR @.-1
52 00320 002614 .XBSP
53       006320 BSPNS=JSR @.-1
54       006207 MESSAGE=JSR @IMES?S
55       006215 TYPZO=JSR @IZOC?T
56       006216 OCTIN=JSR @ITI?O
57       006222 ASCIN=JSR @ITI?A
58       006214 TYPEO=JSR @IPOC?T
59       006223 CALIB=JSR @ICAL?IB
60       006212 TYPZD=JSR @IPDE?C
```

```
 0011 MT110
01       006211 TYPEA=JSR @ITYP?E
02       006212 TYPED=JSR @IPDE?C
03
04       000121 ASCR=C15
05       000123 CONO=C17
```

```
'0012 MT110
01       000500          .LOC    500
02
03 00500 000403 BEGIN:   JMP     DIAG            ; START DIAGNOSTIC HERE
04 00501 002311          JMP @IWLT               ;WRITE LOCK TEST
05 00502 002312          JMP @IETT               ;END TAPE TEST
06
07 00503 062677 DIAG:    IORST                   ; CLEAR ALL I/O
08 00504 030045          LDA     2,NEST          ; GET THE POINTER
09 00505 021000          LDA     0,0,2           ; GET I/O MODULE FLAG
10 00506 101005          MOV     0,0,SNR         ; TEST FOR FLAG SET
11 00507 000411          JMP     DIAG1           ; IF NOT SET, SET POINTER
12 00510 021001          LDA     0,1,2           ; GET EGGS + 1
13 00511 024135          LDA     1,C77           ; GET MASK BITS
14 00512 124000          COM     1,1             ; SET TO MASK INSTRUCTION
15 00513 123525          ANDZL   1,0,SNR         ; ISOLATE I/O MOD
16 00514 000404          JMP     DIAG1           ; IF NOT SET, CLEAR POINTER
17 00515 025004          LDA     1,4,2           ; GET TOP OF MEMORY
18 00516 106400          SUB     0,1             ; SUBTRACT MOD FROM TOP
19 00517 121400          INC     1,0             ; MEMORY TOP + 1
20 00520 040221 DIAG1:   STA     0,IOM?0D        ; SET I/O MOD POINTER
21
22               ; START DIAGNOSTIC HERE
23
24 00521 102520          SUBZL   0,0             ; SET REGISTER TO 1
25 00522 042231          STA     0,@IPC7?        ; SET TAB TO ONE SPACE
26 00523 102400          SUB     0,0             ; CLEAR REGISTER
27 00524 042205          STA     0,@ISWR?EG      ; INITIALIZE SWREG
28
29                       ;CHECK SELB BUS LINE
30
31 00525 062677 A0:      IORST                   ; CLEAR I/O SYSTEM
32 00526 063500          SKPBZ   0               ; TEST FOR SELB LINE SET
33 00527 000006          JMP     BHALT           ; IF SET, IRRECOVERABLE ERROR
34 00530 006264          JSR     @INTIL          ; GO INITIALIZE SYSTEM
35
36                       ;CHECK SELD BUS LINE
37
38 00531 006276 A1:      JSR @SETP               ;DO SKPDZ, DEVICE 0
39 00532 062677          IORST                   ; CLEAR I/O SYSTEM
40 00533 006224          WAIT                    ; WAIT FOR CONSOLE INPUT
41 00534 000067          .200MS
42 00535 152520          SUBZL   2,2             ; SET = 1 FOR ERROR PRINT
43 00536 063700          SKPDZ   0               ; TEST FOR SELD LINE SET
44 00537 006314          HALT1                   ; IF SET, GO PROCESS ERROR
45 00540 006277          JSR @LOOP.
46
47                       ;CHECK MTA BUSY FLOP OFF
48
49 00541 006276 A2:      JSR     @SETP           ;DO BUSY TEST ON
50 00542 030106          LDA     2,C2            ; SET = 2 FOR ERROR PRINT
51 00543 063522          SKPBZ   .MTA            ;MTA
52 00544 006314          HALT1                   ;CANNOT RESET MT BUSY
53 00545 006277          JSR     @LOOP.
54
55                       ;CHECK MTA DONE FLOP OFF
56
57 00546 006276 A3:      JSR     @SETP           ;DONE TEST ON MTA
58 00547 030111          LDA     2,C5            ; SET = 5 FOR ERROR PRINT
59 00550 063722          SKPDZ   .MTA
60 00551 006314          HALT1                   ;CANNOT RESET MT DONE
```

```
 0013 MT110
01 00552 006277          JSR     @LOOP.
02
03                       ;CHECK MT SELECT,DOB,DIB,ADDR REG.  IN-OUT
04
05 00553 006276 A4:      JSR     @SETP           ;LOAD CA WITH 125252
06 00554 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
07 00555 020101          LDA     0,C125252       ;READ BACK AND CHECK
08 00556 062022          DOB     0,.MTA
09 00557 065422          DIB     1,.MTA
10 00560 101120          MOVZL   0,0
11 00561 101220          MOVZR   0,0             ;AC0=VALUE SENT TO ADDR REG
12 00562 106414          SUB#    0,1,SZR         ;AC1=VALUE READ BACK
13 00563 006314          HALT1                   ;CHECK THE MT SELECT, LOAD ADDRESS
14 00564 006277          JSR     @LOOP.          ;ENABLE ADDR. A&B, AND ADDR. REG.
15
16                       ;CHECK CA REGISTER IN&OUT
17
18 00565 006276 A5:      JSR     @SETP           ;LOAD CA WITH 052525
19 00566 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
20 00567 020102          LDA     0,C52525        ;READ BACK AND CHECK
21 00570 062022          DOB     0,.MTA
22 00571 065422          DIB     1,.MTA
23 00572 106414          SUB#    0,1,SZR         ;AC0=VALUE SENT TO ADDR REG
24 00573 006314          HALT1                   ;AC1=VALUE READ BACK
25 00574 006277          JSR     @LOOP.          ;CHECK ADDR REG DATA PATHS
26
27                       ;CHECK FOR ILLEGAL MT SELECT
28
29 00575 024062 A6:      LDA     1,CMTA          ;LOAD CA WITH 052525
30 00576 044404          STA     1,A6.2          ;READ BACK USING EVERY
31 00577 006275 A6.1:    JSR     @SETP1          ;DEVICE CODE EXCEPT MTA
32 00600 020102          LDA     0,C52525        ;DATA SHOULD NOT BE
33 00601 062022          DOB     0,.MTA          ;FOUND!
34 00602 075402 A6.2:    DIB     3,2             ;THIS IS CHANGED BY PROG
35 00603 030120          LDA     2,C14           ; SET = 14 FOR ERROR PRINT
36 00604 162415          SUB#    3,0,SNR
37 00605 006314          HALT1                   ;AC0=DATA SENT TO ADDR REG
38                                               ;AC1=DATA READ BACK
39                                               ;PROBLEM IS IN MT SELECT
40 00606 006277          JSR     @LOOP.          ;THE DEVICE SELECT GATE.
41 00607 024773          LDA     1,A6.2
42 00610 125400 A6.3:    INC     1,1
43 00611 020135          LDA     0,C77
44 00612 123405          AND     1,0,SNR
45 00613 000406          JMP     A7              ;DONE ALL GO ON
46 00614 030243          LDA     2,DEVICE
47 00615 142415          SUB#    2,0,SNR
48 00616 000772          JMP     A6.3            ;MTA, INC AGAIN
49 00617 044763          STA     1,A6.2
50 00620 000757          JMP     A6.1
51
52                       ;CHECK CA REGISTER, ALL DATA PATTERNS
53
54 00621 102400 A7:      SUB     0,0             ;LOAD CA, READ BACK
55 00622 006275          JSR     @SETP1          ;AND CHECK.  USE
56 00623 062022          DOB     0,.MTA          ;ALL FROM 0-077777
57 00624 065422          DIB     1,.MTA
58 00625 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
59 00626 106414          SUB#    0,1,SZR         ;AC0=DATA SENT TO ADDR REG
60 00627 006314          HALT1                   ;AC1=DATA READ BACK
```

```
 0014 MT110
01 00630 006277          JSR     @LOOP.          ;ADDR REG IS PATTERN SENSITIVE
02 00631 101400          INC     0,0
03 00632 101113          MOVL#   0,0,SNC
04 00633 000767          JMP     A7+1
05 00634 000401          JMP     A8
06
07                       ;CHECK IO RESET OF CA REGISTER
08
09 00635 006276 A8:      JSR     @SETP           ;LOAD CA WITH ALL 1'S
10 00636 102000          ADC     0,0             ;DO IORST,  READ
11 00637 062022          DOB     0,.MTA          ;CA AND CHECK FOR 0'S
12 00640 062677          IORST
13 00641 006224          WAIT                    ; WAIT FOR CONSOLE INPUT
14 00642 000067          .200MS
15 00643 030121          LDA     2,C15           ; SET = 15 FOR ERROR PRINT
16 00644 065422          DIB     1,.MTA          ;AC0=DATA SENT TO ADDR REG
17 00645 125004          MOV     1,1,SZR         ;AC1=DATA READ BACK
18 00646 006314          HALT1                   ;CHECK IORST, RESET, AND
19 00647 006277          JSR     @LOOP.          ;ADDRESS REGISTER RESET GATES
20
21                       ;CHECK CA LOAD
22
23 00650 006276 A8A:     JSR     @SETP           ;LOAD CA WITH 010421
24 00651 102400          SUB     0,0             ;LOAD CA WITH ALL ZEROS
25 00652 024071          LDA     1,C10421        ;READ AND CHECK
26 00653 066022          DOB     1,.MTA
27 00654 062022          DOB     0,.MTA
28 00655 065422          DIB     1,.MTA
29 00656 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
30 00657 125004          MOV     1,1,SZR         ;CA LOADING ERROR
31 00660 006314          HALT1                   ;AC1=(CA)
32 00661 006277          JSR     @LOOP.          ;(CA) SHOULD BE ZERO
33
34 00662 006276 A8B:     JSR     @SETP           ;LOAD CA WITH 21042
35 00663 102400          SUB     0,0             ;LOAD CA WITH ALL ZEROS
36 00664 024072          LDA     1,C21042        ;READ AND CHECK
37 00665 066022          DOB     1,.MTA
38 00666 062022          DOB     0,.MTA
39 00667 065422          DIB     1,.MTA
40 00670 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
41 00671 125004          MOV     1,1,SZR         ;CA LOADING ERROR
42 00672 006314          HALT1                   ;AC1=(CA)
43 00673 006277          JSR     @LOOP.          ;(CA) SHOULD BE ZERO
44
45 00674 006276 A8C:     JSR @SETP               ;LOAD CA WITH 42104
46 00675 102400          SUB     0,0             ;LOAD CA WITH ALL ZEROS
47 00676 024073          LDA     1,C42104
48 00677 066022          DOB     1,.MTA
49 00700 062022          DOB     0,.MTA
50 00701 065422          DIB     1,.MTA
51 00702 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
52 00703 125004          MOV     1,1,SZR         ;CA LOADING ERROR
53 00704 006314          HALT1                   ;AC1=(CA)
54 00705 006277          JSR     @LOOP.          ;(CA) SHOULD BE ZERO
55
56 00706 006276 A8D:     JSR     @SETP           ;LOAD CA WITH 4210
57 00707 102400          SUB     0,0             ;LOAD CA WITH ALL ZEROS
58 00710 024074          LDA     1,C421T         ;READ AND CHECK
59 00711 066022          DOB     1,.MTA
60 00712 062022          DOB     0,.MTA
```

```
 0015 MT110
01 00713 065422          DIB     1,.MTA
02 00714 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
03 00715 125004          MOV     1,1,SZR         ;CA LOADING ERROR
04 00716 006314          HALT1                   ;AC1=(CA)
05 00717 006277          JSR     @LOOP.          ;(CA) SHOULD BE ZERO
06
07                       ;CHECK CA LOAD
08
09 00720 006276 A8E:     JSR     @SETP
10 00721 024071          LDA     1,C10421
11 00722 034072          LDA     3,C21042
12 00723 030073          LDA     2,C42104
13 00724 020074          LDA     0,C421T
14 00725 066022          DOB     1,.MTA
15 00726 076022          DOB     3,.MTA
16 00727 072022          DOB     2,.MTA
17 00730 062022          DOB     0,.MTA
18 00731 065422          DIB     1,.MTA
19 00732 030122          LDA     2,C16           ; SET = 16 FOR ERROR PRINT
20 00733 106414          SUB#    0,1,SZR         ;CA LOADING ERROR
21 00734 006314          HALT1                   ;AC1=(CA)
22 00735 006277          JSR     @LOOP.          ;(CA) SHOULD BE 004210
23
24                       ;CHECK FOR "TUR"
25
26 00736 006276 A9:      JSR     @SETP           ;FIRST ATTEMPT TO
27 00737 006306          SELECT                  ;READ STATUS
28 00740 126520          SUBZL   1,1             ; SET READY BIT
29 00741 060422          DIA     0,.MTA          ;TAPE UNIT READY (TUR)
30 00742 107415          AND#    0,1,SNR         ; IS BIT 15.
31 00743 006300          EHALT                   ;CHECK DRIVE FOR THE
32 00744 006277          JSR     @LOOP.          ;READY STATE.
33
34                       ;CHECK MT START AND BUSY FLOP
35
36 00745 006276 A9A:     JSR     @SETP           ;SEND START PULSE
37 00746 006306          SELECT
38 00747 030111          LDA     2,C5            ; SET = 5 FOR ERROR PRINT
39 00750 060122          NIOS    .MTA            ;SKIP ON BUSY SET
40 00751 063622          SKPDN   .MTA            ;MT START DID NOT CLEAR DONE
41 00752 000403          JMP     A9B             ; IF CLEAR, TEST BUSY
42 00753 006314          HALT1
43 00754 000404          JMP     A9C             ; TEST LOOP ON ERROR
44 00755 030110 A9B:     LDA     2,C4            ; SET = 4 FOR ERROR PRINT
45 00756 063422          SKPBN   .MTA            ;MT START DID NOT SET BUSY
46 00757 006314          HALT1                   ;CHECK MT START,BUSY,DONE FLOPS
47 00760 006277 A9C:     JSR     @LOOP.          ;AND SET LINE
48
49                       ;CHECK IORST OF BUSY FLOP (CLEAR COM)
50
51 00761 006276 A10:     JSR     @SETP           ;SEND START PULSE
52 00762 006306          SELECT
53 00763 060122          NIOS    .MTA
54 00764 062677          IORST
55 00765 006224          WAIT                    ; WAIT FOR CONSOLE INPUT
56 00766 000067          .200MS
57 00767 030107          LDA     2,C3            ; SET = 3 FOR ERROR PRINT
58 00770 063522          SKPBZ   .MTA            ;IORST DIDN'T RESET BUSY
59 00771 006314          HALT1                   ;CHECK CLEAR COM AND
60 00772 006277          JSR     @LOOP.          ;MT BUSY RESET LINE
```

```
 0016 MT110
01
02                      ;CHECK CLEAR PULSE
03
04 00773 006276 A11:    JSR     @SETP           ;SEND START PULSE
05 00774 006306         SELECT
06 00775 060122         NIOS    .MTA
07 00776 060222         NIOC    .MTA
08 00777 030111         LDA     2,C5            ; SET = 5 FOR ERROR PRINT
09 01000 063622         SKPDN   .MTA            ;MT CLR DIDN'T RESET DONE
10 01001 000403         JMP     A11A            ; IF RESET, TEST BUSY
11 01002 006314         HALT1                   ; IF SET, PROCESS ERROR
12 01003 000404         JMP     A11B            ; GO LOOP ON ERROR
13 01004 030106 A11A:   LDA     2,C2            ; SET = 2 FOR ERROR PRINT
14 01005 063522         SKPBZ   .MTA
15 01006 006314         HALT1                   ;MT CLR DIDN'T RESET BUSY
16 01007 006277 A11B:   JSR     @LOOP.          ;CHECK MT CLR BUSY,DONE FLOPS
17
18                      ;CHECK DOA-MTLC-COMMAND DECODER
19
20 01010 006275 A12:    JSR     @SETP1          ;SELECT REWIND
21 01011 006306         SELECT                  ;SEND START PULSE
22 01012 030106         LDA     2,C2            ; SET = 2 FOR ERROR PRINT
23 01013 020233         LDA     0,C1X           ;MAKE SURE BUSY DOESN'T SET
24 01014 061122         DOAS    0,.MTA          ;CHECK E29 GATE DRIVING THE
25 01015 063522         SKPBZ   .MTA            ;MT BUSY SET LINE.  REWIND IS ON
26 01016 006314         HALT1                   ;AND MT START IS SETTING BUSY.
27 01017 006277         JSR     @LOOP.          ;CHECK COMMAND DECODER, E2.
28
29                      ;CHECK DIA AND STATUS REGISTER AT BOT
30
31 01020 006275 A13:    JSR     @SETP1          ;REWIND TAPE
32 01021 006306         SELECT                  ;READ AND CHECK STATUS
33 01022 006317         REWIND
34 01023 006265         STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD.
35 01024 100201         100201
36 01025 006300         EHALT                   ;CHECK RD STATUS, REWIND THRU TO
37 01026 000401         JMP     .+1             ;TAPE, AND BOT LINE FORM TAPE.
38 01027 006277         JSR     @LOOP.          ;ALSO BOT STATUS REG.  GATE
39
40                      ;CHECK UNIT SELECT LOGIC, ERASE COMMAND DECODE
41
42 01030 020134 A14:    LDA     0,C70           ;SELECT EACH UNIT
43 01031 040232         STA     0,X70           ;IN SEQUENCE 0-7
44 01032 006276 A14.1:  JSR     @SETP           ;CHECK FOR TUR IF
45 01033 020232         LDA     0,X70           ;UNIT IS SELECTED.
46 01034 061022         DOA     0,.MTA          ;OTHERWISE CHECK FOR
47 01035 006263         JSR     @ICKST          ;NO TUR.
48 01036 006300         EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
49                                              ;AC2 13-15=UNIT JUST TESTED
50                                              ;IF UNIT TESTED=UNIT SELECTED
51                                              ;PROCEED ON THE STATUS INFO.
52 01037 006277         JSR     @LOOP.          ;IF NOT CHECK FOR ERRONEOUS TUR.
53
54 01040 010232         ISZ     X70
55 01041 020232         LDA     0,X70
56 01042 024136         LDA     1,C100
57 01043 122414         SUB#    1,0,SZR
58 01044 000766         JMP     A14.1
59
60                      ;CHECK FOR "TUR" FOLLOWING ERASE
```

```
 0017 MT110
01
02 01045 006275 A14A:   JSR     @SETP1          ;REWIND TO START TEST
03 01046 006317         REWIND                  ;AT BOT
04 01047 006266         ERASE                   ;ERASE & WAIT 1 SEC
05 01050 006265         STATUS
06 01051 000001         1                       ;AC1=EXPECTED STATUS
07 01052 006300         EHALT                   ;AC0=ACTUAL STATUS
08 01053 000401         JMP     .+1             ;FOLLOWING ERASE
09 01054 006277         JSR     @LOOP.
10
11                      ;CHECK FOR "DONE" FOLLOWING ERASE
12
13 01055 006275 A14B:   JSR     @SETP1          ;REWIND TO START TEST
14 01056 006317         REWIND                  ;AT BOT
15 01057 006266         ERASE                   ;ERASE & TIME 1 SEC
16 01060 030106         LDA     2,C2            ; SET = 6 FOR ERROR PRINT
17 01061 063422         SKPBN   .MTA
18 01062 000403         JMP     A14C            ; IF NOT BUSY, TEST DONE
19 01063 006314         HALT1                   ; IF SET, PROCESS ERROR
20 01064 000404         JMP     A14D            ; TEST FOR LOOP ON ERROR
21 01065 030112 A14C:   LDA     2,C6            ; SET = 6 FOR ERROR PRINT
22 01066 063622         SKPDN   .MTA            ;BUSY=1 OR DONE=0
23 01067 006314         HALT1                   ;FOLLOWING ERASE
24 01070 006277 A14D:   JSR     @LOOP.
25
26                      ;CHECK REWIND-ERASE-REWIND SEQUENCE
27
28 01071 006275 A15:    JSR     @SETP1          ;REWIND WAIT 5 SEC.
29 01072 006317         REWIND                  ;ERASE WAIT 1 SEC.
30 01073 006266         ERASE                   ;CHECK STATUS (TUR)
31 01074 006265         STATUS                  ;REWIND-CHK STATUS
32 01075 000001         1
33 01076 006300         EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
34 01077 000406         JMP     A15E            ;IF (BOT), ERASE DIDN'T MOVE TAPE.
35                                              ;CHECK ERASE COMMAND THRU TO TAPE
36 01100 006317         REWIND
37 01101 006265         STATUS
38 01102 100201         100201                  ;AC1=EXPECTED STATUS, AC0=BAD.
39 01103 006300         EHALT                   ;REWIND FOLLOWING ERASE DIDN'T
40 01104 000401         JMP     .+1             ;WORK.  THE CLUE IS IN THE BAD
41 01105 006277 A15E:   JSR     @LOOP.          ;STATUS.
42
43                      ;CHECK REWINDING STATUS BIT
44
45 01106 006275 A16:    JSR     @SETP1          ;DO ERASE-RWNS
46 01107 006266         ERASE                   ;CHECK STATUS
47 01110 006316         RWNS                    ;DURING REWIND
48 01111 024151         LDA     1,C20K
49 01112 060422         DIA     0,.MTA
50 01113 123405         AND     1,0,SNR
51 01114 006300         EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
52 01115 006317         REWIND                  ;CHECK REWINDING STATUS BIT GATE
53 01116 006277         JSR     @LOOP.
54
55
56                      ;CHECK SIMPLE 4 CHAR WRITE
57
58 01117 006276 A20:    JSR     @SETP           ;WRITE 2WD
59 01120 006270         GEN                     ;CHECK CA, DONE FLAG
60 01121 000377 A20.1:  377
```

```
 0018 MT110
01 01122 006267         WRITE
02 01123 000102         102
03 01124 030112         LDA     2,C6            ; SET = 6 FOR ERROR PRINT
04 01125 063722         SKPDZ   .MTA            ;DONE FLAG SHOULD BE ON.
05 01126 000403         JMP     .+3             ;THIS IS THE FIRST COMPLETE WRITE
06 01127 006314         HALT1                   ;AND INVOLVES ALL THE WRITE
07 01130 000416         JMP     A20E            ;LOGIC.  TO COMPLETE A WRITE
08                                              ;ERF-LOAD DELAY-DLY OVER-STOP-
09 01131 024063         LDA     1,OBUFF         ;TUR-MT DONE, CLEAR MT BUSY.
10 01132 030106         LDA     2,C2            ;CHECK WC OVER, INCREMENT WC.
11 01133 147000         ADD     2,1
12 01134 061422         DIB     0,.MTA
13 01135 030126         LDA     2,C21           ; SET = 21 FOR ERROR PRINT
14 01136 106415         SUB#    0,1,SNR
15 01137 000403         JMP     .+3             ;ADDR REG. WRONG!  AC1=GOOD
16 01140 006314         HALT1                   ;AC0=BAD.  TWO OUTPUT XFERS
17 01141 000405         JMP     A20E            ;SHOULD HAVE OCCURRED
18 01142 006265         STATUS
19 01143 000001         1
20 01144 006300         EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
21 01145 000401         JMP     .+1             ;DATA PATTERN IS 1-0-1-0
22                                              ;FOR EACH TRACK.  LATERAL
23 01146 006277 A20E:   JSR     @LOOP.          ;PARITY IS ALWAYS 1.
24
25                      ;CHECK INTERRUPT AND DISABLE
26
27 01147 006275 A21:    JSR     @SETP1          ;WRITE 2 WDS
28 01150 006267         WRITE
29 01151 000102         102
30 01152 030116         LDA     2,C12           ; SET = 12 FOR ERROR PRINT
31 01153 060177         INTEN                   ;ENABLE INTERRUPT
32 01154 000401         JMP     .+1             ;ALLOW 1 INST BEFORE INT.
33 01155 006314         HALT1                   ;NO INTERRUPT.  CHECK MT INT REQ.
34 01156 006277         JSR     @LOOP.          ;INTR GATE, INTP IN GATE.
35
36                      ;CHECK PROPER RESPONSE TO INTA
37
38 01157 006275 A21A:   JSR     @SETP1          ;WRITE 2 WORDS
39 01160 006267         WRITE
40 01161 000102         102
41 01162 030124         LDA     2,C20           ; SET = 20 FOR ERROR PRINT
42 01163 061477         INTA    0               ;READ DEVICE ADDR
43 01164 024243         LDA     1,DEVICE
44 01165 122414         SUB#    1,0,SZR         ;AC1=CORRECT INTA RESPONSE
45 01166 006314         HALT1                   ;AC0=BAD
46 01167 006277         JSR     @LOOP.
47
48                      ;INSURE THAT MSKO WILL DISABLE INTERRUPT
49
50 01170 006275 A21B:   JSR     @SETP1          ;WRITE 2 WORDS
51 01171 006267         WRITE
52 01172 000102         102
53 01173 024131         LDA     1,C40
54 01174 066077         MSKO    1               ;DISABLE MTA INTERRUPTS
55 01175 030117         LDA     2,C13           ; SET = 13 FOR ERROR PRINT
56 01176 060177         INTEN                   ;ALLOW INTERRUPTS
57 01177 000401         JMP     .+1
58 01200 000402         JMP     .+2             ;NOT PERFORMED IF INTERRUPT
59 01201 006314         HALT1                   ;ILLEGAL INTERRUPT.  CHECK
60 01202 060277         INTDS                   ;MT INT DISABLE FLOP,
```

```
 0019 MT110
01 01203 006277         JSR     @LOOP.          ;AMD INPUTS
02                      ;WRITE 16 CHAR RECORD.  ODD PARITY
03                      ;PATTERN IN A22.2
04
05 01204 102000 A22:    ADC     0,0
06 01205 040405         STA     0,A22.2
07 01206 020246         LDA     0,POINT
08 01207 040171         STA     0,IIDX1
09 01210 006275 A22.1:  JSR     @SETP1          ;WRITE
10                                              ;CHECK DONE, CA
11 01211 006270         GEN                     ;STATUS.
12 01212 177777 A22.2:  177777
13 01213 006267         WRITE
14 01214 000110         110
15 01215 030112         LDA     2,C6            ; SET = 6 FOR ERROR PRINT
16 01216 063722         SKPDZ   .MTA            ;NO DONE FLAG!  RECORDS TO NOW
17 01217 000403         JMP     .+3             ;HAVE BEEN 4 CHAR.  THIS ONE IS
18 01220 006314         HALT1                   ;16 CHAR.  THIS IS THE ONLY
19 01221 000416         JMP     A22E            ;DIFFERENCE.  READ AFTER WRITE
20                                              ;DETECTS ERF,AND ERF-LOAD DELAY-
21 01222 024063         LDA     1,OBUFF         ;DLY OVER-STOP-TUR-MT DONE.
22 01223 030114         LDA     2,C10           ;CHECK WC OVER, INCREMENT WC
23 01224 147000         ADD     2,1
24 01225 030126         LDA     2,C21           ; SET = 21 FOR ERROR PRINT
25 01226 061422         DIB     0,.MTA
26 01227 106415         SUB#    0,1,SNR         ;CA WRONG.  AC1=CORRECT
27 01230 000403         JMP     .+3             ;AC0=BAD.  CHECK DATA
28 01231 006314         HALT1                   ;CHAN CONTROL AND INCREMENT
29 01232 000405         JMP     A22E            ;WC.  10 OCTAL WORDS WRITTEN
30
31 01233 006265         STATUS
32 01234 000001         1
33 01235 006300         EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
34 01236 000401         JMP     .+1             ;THE CLUE IS IN THE BAD STATUS
35 01237 006277 A22E:   JSR     @LOOP.
36
37 01240 020171         LDA     0,IIDX1
38 01241 024254         LDA     1,LST
39 01242 106405         SUB     0,1,SNR
40 01243 000405         JMP     A23             ;GO TO NEXT TEST
41 01244 010171         ISZ     IIDX1           ;%
42 01245 022171         LDA     0,@IIDX1
43 01246 040744         STA     0,A22.2         ;SET NEW DATA PATTERN
44 01247 000741         JMP     A22.1
45                      ;WRITE 16 CHAR RECORD.  EVEN PARITY
46                      ;PATTERN IS IN A23.2
47
48 01250 102000 A23:    ADC     0,0
49 01251 040405         STA     0,A23.2
50 01252 020255         LDA     0,PONTR
51 01253 040171         STA     0,IIDX1
52 01254 006275 A23.1:  JSR     @SETP1          ;WRITE
53                                              ;CHECK DONE, CA
54 01255 006270         GEN                     ;STATUS
55 01256 177777 A23.2:  177777
56 01257 006267         WRITE
57 01260 100110         100110
58 01261 030112         LDA     2,C6            ; SET = 6 FOR ERROR PRINT
59 01262 063722         SKPDZ   .MTA            ;NO DONE FLAG!  THIS HAS WORKED
60 01263 000403         JMP     .+3             ;MANY TIMES BEFORE IN ODD PAR.
```

```
 0020 MT110
01 01264 006314         HALT1                   ;CHECK READ AFTER WRITE ERF
02 01265 000416         JMP     A23E            ;DETECTION.
03 01266 024063         LDA     1,OBUFF
04 01267 030114         LDA     2,C10
05 01270 147000         ADD     2,1
06 01271 030126         LDA     2,C21           ; SET = 21 FOR ERROR PRINT
07 01272 061422         DIB     0,.MTA
08 01273 106415         SUB#    0,1,SNR         ;ADDR. REG. WRONG.  AC1=CORRECT
09 01274 000402         JMP     .+3             ;AC0=BAD.  CHECK DATA CHAN REQ.
10 01275 006314         HALT1                   ;AND INCREMENT WC.
11 01276 000405         JMP     A23E
12 01277 006265         STATUS
13 01300 000001         1
14 01301 006300         EHALT                   ;AC1=CORRECT STATUS, AC0=BAD
15 01302 000401         JMP     .+1             ;PARITY ERRORS ARE LIKELY DUE
16                                              ;TO THE PARITY CHECKING LOGIC
17 01303 006277 A23E:   JSR     @LOOP.          ;(EVEN PARITY)
18
19 01304 020171         LDA     0,IIDX1
20 01305 024262         LDA     1,LST1
21 01306 106405         SUB     0,1,SNR
22 01307 000405         JMP     A24
23 01310 010171         ISZ     IIDX1           ;%
24 01311 022171         LDA     0,@IIDX1
25 01312 040744         STA     0,A23.2
26 01313 000741         JMP     A23.1
27
28                      ;TEST ILLEGAL STATUS
29
30 01314 034234 A24:    LDA     3,CX            ;GIVE START TO UNSELECTED
31 01315 030105         LDA     2,C1            ; TAPE
32 01316 175400         INC     3,3             ;AC2=UNIT
33 01317 173400         AND     3,2
34 01320 006276         JSR     @SETP
35 01321 071022         DOA     2,.MTA          ;START READ
36 01322 060122         NIOS    .MTA
37 01323 063522         SKPBZ   .MTA
38 01324 000777         JMP     .-1
39 01325 060422         DIA     0,.MTA          ;READ STATUS
40 01326 024100         LDA     1,C110000       ;AC1=THE 2 BITS EXPECTED
41 01327 123400         AND     1,0             ;AC0=THE 2 BITS FOUND
42 01330 122414         SUB#    1,0,SZR         ;LOOKING FOR ERROR AND ILLEGAL
43 01331 006300         EHALT                   ;STATUS.  CHECK ILLEGAL AND SET
44                                              ;GATES (E31), ALSO STATUS BIT
45 01332 006277         JSR     @LOOP.          ;GATE (E52).
46                      ;TEST ILLEGAL STATUS
47
48 01333 006275 A25:    JSR     @SETP1          ;REWIND.  GIVE
49 01334 006317         REWIND                  ;SPACE BACK AT BOT
50 01335 006272         BSPACE
51 01336 000001         1
52 01337 006265         STATUS
53 01340 110201         110201                  ;AC1=EXPECTED STATUS, AC0=BAD.
54 01341 006300         EHALT                   ;CHECK SPACE BKD&BOT GATE ON
55 01342 000401         JMP     .+1             ;ILLEGAL (E31).  CHECK SPACE BKD
56 01343 006277         JSR     @LOOP.          ;COMMAND DECODE
57
58                      ;CHECK "DONE" FOLLOWING WRITE EOF
59
59 01344 006275 A26A:   JSR     @SETP1
```

```
 0021 MT110
01 01345 006271          WEOF                    ;WRITE EOF
02 01346 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
03 01347 063622          SKPDN   .MTA
04 01350 006314          HALT1                   ;NO DONE FLAG
05 01351 006277          JSR     @LOOP.          ;ON WRITE EOF
06
07                       ;CHECK WRITE EOF STATUS
08
09 01352 006276 A26:     JSR     @SETP           ; WEOF
10 01353 006271          WEOF                    ;AND CHECK STATUS
11 01354 006265          STATUS
12 01355 100401          100401
13 01356 006300          EHALT                   ;AC1=EXPECTED STATUS.  AC0=BAD
14 01357 000401          JMP     .+1             ;CHECKING FOR EOF,ERROR.   CHECK EOF
15 01360 006277          JSR     @LOOP.          ;FLOP AND C-D INPUTS.
16
17
18 01361 006275 A28:     JSR     @SETP1          ;REWIND-WRITE 1 REC.
19 01362 006317          REWIND                  ;BACKSPACE AND CHECK
20 01363 006267          WRITE                   ;DONE AND CORRECT
21 01364 000102          102                     ;STATUS
22 01365 006272          BSPACE
23 01366 000001          1
24 01367 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
25 01370 063722          SKPDZ   .MTA            ;NO DONE FLAG FOLLOWING
26 01371 000403          JMP     .+3             ;BACKSPACE.   CHECK SPACE BKD
27 01372 006314          HALT1                   ;COMMAND DECODE AND DRIVER
28 01373 000405          JMP     A28E            ;TO TAPE (E20)
29 01374 006265          STATUS
30 01375 000001          1
31 01376 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
32 01377 000401          JMP     .+1
33 01400 006277 A28E:    JSR     @LOOP.
34
35                       ;CHECK BACKSPACE
36
37 01401 006275 A30:     JSR     @SETP1          ;REWIND-CHECK FOR BOT
38 01402 006317          REWIND                  ;WRITE 2 REC.   CHECK
39 01403 006265          STATUS                  ;STATUS, BACKSPACE 2
40 01404 100201          100201                  ;CHECK STATUS AT BOT
41 01405 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD.
42 01406 000416          JMP     A30E            ;LOOKING FOR EOT FOLLOWING
43                                               ;REWIND.   THIS SHOULD WORK!
44 01407 006267          WRITE
45 01410 000202          202
46 01411 006272          BSPACE
47 01412 000002          2
48 01413 006265          STATUS
49 01414 000001          1
50 01415 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD.
51 01416 000406          JMP     A30E
52 01417 061422          DIB     0,.MTA
53 01420 024140          LDA     1,C102
54 01421 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
55 01422 122414          SUB#    1,0,SZR         ;ADDRESS REG.  ERROR.
56 01423 006314          HALT1                   ;2 REC SHOULD ICR IT
57 01424 006277 A30E:    JSR     @LOOP.          ;TWICE DURING BACKSPC
58                       ;CHECK BACKSPACE
59
60 01425 006275 A31.     JSR     @SETP1          ;REWIND-WRITE 3 REC.
```

```
 0022 MT110
01 01426 006317          REWIND                  ;BACKSPACE 4, CHECK
02 01427 006267          WRITE                   ;BOT AND CA FOR
03 01430 000302          302                     ;3 COUNTS
04 01431 006272          BSPACE
05 01432 000004          4
06 01433 006265          STATUS
07 01434 100201          100201
08 01435 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
09                                               ;BOT SHOULD STOP OPERATION
10                                               ;IN 3 BACKSPACES
11 01436 006277          JSR     @LOOP.
12
13                       ;CHECK SPACE FORWARD
14
15 01437 006275 A32:     JSR     @SETP1          ;REWIND-WRITE 3 REC.
16 01440 006317          REWIND                  ;REWIND-CHECK STATUS
17 01441 006267          WRITE                   ;SPACE FORWARD-CHECK
18 01442 000302          302                     ;FOR NO BOT, & PROPER
19 01443 006317          REWIND                  ;ADDR. REG.
20 01444 006265          STATUS
21 01445 100201          100201
22 01446 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
23 01447 000414          JMP     A32E            ;THIS HAS WORKED MANY TIMES BEFORE
24 01450 006305          SPACE
25 01451 000001          1
26 01452 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD.
27 01453 000001          1                       ;THIS IS THE FIRST SAPCE FORWARD.
28 01454 006300          EHALT                   ;CHECK COMMAND DECODER-SPACE FWD-
29 01455 000406          JMP     A32E            ;COUNT WC (E43)
30 01456 024137          LDA     1,C101
31 01457 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
32 01460 061422          DIB     0,.MTA          ;ADDR REG WRONG.  AC1=CORRECT
33 01461 106414          SUB#    0,1,SZR         ;AC0=ACTUAL.  ONE INCR FROM 100
34 01462 006314          HALT1                   ;TO 101 SHOULD HAVE OCCURRED.
35                                               ;CHECK SPACE FWD NOT-COUNT WC-
36 01463 006277 A32E:    JSR     @LOOP.          ;INCREMENT WC.
37
38                       ;CHECK READ INSTRUCTION.  READ 16 CHAR
39                       ;RECORDS.  PATTERN IN A33.2 (ODD PARITY)
40
41 01464 102000 A33:     ADC     0,0             ;REWIND-WRITE 1 REC.
42 01465 040406          STA     0,A33.2         ;CHECK STATUS-REWIND-
43 01466 020246          LDA     0,POINT         ;CHECK STATUS-READ-
44 01467 040171          STA     0,IIDX1         ;CHECK DONE,STATUS,
45 01470 006275 A33.1:   JSR     @SETP1          ;ADDR. REG., & DATA
46 01471 006317          REWIND                  ;BUFFER
47 01472 006270          GEN
48 01473 017777 A33.2:   17777
49 01474 006267          WRITE
50 01475 000110          110
51 01476 006265          STATUS
52 01477 000001          1
53 01500 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
54 01501 000436          JMP     A33E            ;THIS MUCH HAS WORKED BEFORE,
55 01502 006317          REWIND
56 01503 006265          STATUS
57 01504 100201          100201                  ;AC1=EXPECTED STATUS, AC0=BAD
58 01505 006300          EHALT                   ;LOOKING FOR BOT BEFORE TRYING
59 01506 000431          JMP     A33E            ;FIRST READ.  THIS HAS WORKED BEFORE
60 01507 006301          CLEAR
```

```
 0023 MT110
01 01510 006301          READ
02 01511 000110          110
03 01512 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
04 01513 063722          SKPDZ   .MTA            ;NO DONE FLAG FOLLOWING READ.
05 01514 000403          JMP     .+3             ;THIS IS THE FIRST READ INST.
06 01515 006314          HALT1                   ;CHECK READ-DCHI-INCREMENT WC
07 01516 000421          JMP     A33E
08 01517 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
09 01520 000001          1                       ;90% OF THE READ LOGIC HAS BEEN
10 01521 006300          EHALT                   ;CHECKED DURING WRITE OPERATIONS.
11 01522 000415          JMP     A33E            ;PROBLEMS HERE ARE LIKELY IN THE
12                                               ;PROCESSOR INTERFACE AREA.
13 01523 020064          LDA     0,IBUFF
14 01524 024114          LDA     1,C10
15 01525 107000          ADD     0,1
16 01526 061422          DIB     0,.MTA
17 01527 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
18 01530 106415          SUB#    0,1,SNR         ;ADDRESS REGISTER IS WRONG.
19 01531 000403          JMP     .+3             ;AC1=CORRECT, AC0=ACTUAL.
20 01532 006314          HALT1                   ;10 WORD REC READ.  CHECK
21 01533 000404          JMP     A33E            ;MT DCH SEL-DCHM0-INCREMENT WC
22 01534 006303          CHECK                   ;INPUT BUFFER CONTAINS WRONG
23 01535 000010          10                      ;DATA.  INPUT BUFFER SHOULD =
24 01536 006315          HALT2                   ;OUTPUT BUFFER FOR THE FIRST 8
25                                               ;WORDS.  AC0=GOOD WORD, AC1=BAD
26 01537 006277 A33E:    JSR     @LOOP.          ;AC2=ADDRESS OF INPUT BUFFER
27
28 01540 020171          LDA     0,IIDX1
29 01541 024254          LDA     1,LST
30 01542 106405          SUB     0,1,SNR
31 01543 000405          JMP     A34
32 01544 010171          ISZ     IIDX1           ;%
33 01545 022171          LDA     0,@IIDX1
34 01546 040725          STA     0,A33.2
35 01547 000721          JMP     A33.1
36
37                       ;CHECK READ INSTRUCTION.  READ 16 CHAR
38                       ;RECORDS.  PATTERN IN A34.2 (EVEN PARITY)
39
40 01550 102000 A34:     ADC     0,0             ;REWIND-WRITE 1 REC.
41 01551 040406          STA     0,A34.2         ;CHECK STATUS-REWIND
42 01552 020255          LDA     0,PONTR         ;CHECK STATUS-READ
43 01553 040171          STA     0,IIDX1         ;CHECK DONE,STATUS,
44 01554 006275 A34.1:   JSR     @SETP1          ;ADDRESS REG.,DATA IN
45 01555 006217          REWIND                  ;BUFFER
46 01556 006270          GEN
47 01557 177777 A34.2:   177777
48 01560 006267          WRITE
49 01561 100110          100110
50 01562 006265          STATUS
51 01563 000001          1
52 01564 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
53 01565 000436          JMP     A34E            ;THIS MUCH HAS WORKED BEFORE!
54 01566 006317          REWIND
55 01567 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
56 01570 100201          100201                  ;LOOKING FOR BOT STATUS BEFORE
57 01571 006300          EHALT                   ;TRYING FIRST EVEN PARITY READ.
58 01572 000431          JMP     A34E            ;THIS SHOULD WORK!
59 01573 006302          CLEAR
60 01574 006301          READ
```

```
 0024 MT110
01 01575 100110          100110
02 01576 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
03 01577 063722          SKPDZ   .MTA
04 01600 000403          JMP     .+3
05 01601 006314          HALT1                   ;NO DONE FLAG.  CHECK ERF-
06 01602 000421          JMP     A34E            ;DCHI-INCREMENT WC
07 01603 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
08 01604 000001          1                       ;CHECK PARITY CIRCUITS.  THIS
09 01605 006300          EHALT                   ;IS THE ONLY DIFFERENCE FROM
10 01606 000415          JMP     A34E            ;THE LAST TEST.
11 01607 020064          LDA     0,IBUFF
12 01610 024114          LDA     1,C10
13 01611 107000          ADD     0,1
14 01612 061422          DIB     0,.MTA
15 01613 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
16 01614 106415          SUB#    0,1,SNR         ;ADDRESS REG. IS WRONG.  AC1=
17 01615 000403          JMP     .+3             ;CORRECT VALUE, AC0=ACTUAL.
18 01616 006314          HALT1                   ;10 WORD REC. READ.  CHECK MT
19 01617 000404          JMP     A34E            ;DCH SEL-DCHM0-INCREMENT WC
20 01620 006303          CHECK                   ;INPUT BUFFER CONTAINS WRONG
21 01621 000010          10                      ;DATA.  INPUT BUFFER SHOULD
22 01622 006315          HALT2                   ;=OUTPUT BUFFER FOR FIRST 8 WORDS
23                                               ;AC0=GOOD WORD, AC1=BAD WORD
24 01623 006277 A34E:    JSR     @LOOP.          ;AC2=ADDRESS OF THE INPUT BUFFER
25
26 01624 020171          LDA     0,IIDX1
27 01625 024262          LDA     1,LST1
28 01626 106405          SUB     0,1,SNR
29 01627 000405          JMP     A35
30 01630 010171          ISZ     IIDX1           ;%
31 01631 022171          LDA     0,@IIDX1
32 01632 040725          STA     0,A34.2
33 01633 000721          JMP     A34.1
34
35                       ;READ 16 CHAR REC.  WITH WC=63
36
37 01634 006275 A35:     JSR     @SETP1          ;REWIND- WRITE 1 REC.
38 01635 006317          REWIND                  ;CHECK STATUS-REWIND
39 01636 006270          GEN                     ;READ-CHECK DONE,
40 01637 125252          125252                  ;STATUS,ADDR REG,
41 01640 006267          WRITE                   ;AND DATA
42 01641 000110          110
43 01642 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
44 01643 000001          1                       ;DON'T TYR TO READ IT IF IT
45 01644 006300          EHALT                   ;CANNOT BE WRITTEN PROPERLY.
46 01645 000431          JMP     A35E            ;THIS HAS WORKED BEFORE,
47 01646 006317          REWIND
48 01647 006301          READ
49 01650 000177          177
50 01651 030112          LDA     2,C6            ; SET = 6 FOR ERROR PRINT
51 01652 063722          SKPDZ   .MTA
52 01653 000403          JMP     .+3
53 01654 006314          HALT1                   ;NO DONE FLAG.  CHECK INCREMENT
54 01655 000421          JMP     A35E            ;WC-ERF-STOP-TUR
55 01656 006265          STATUS                  ;AC1=EXPECTED STATUS, AC0=BAD
56 01657 000001          1                       ;THIS IS THE FIRST TIME ON A
57 01660 006300          EHALT                   ;READ THAT THE WC DIDN'T MATCH
58 01661 000415          JMP     A35E            ;THE RECORD LENGTH.  CHECK
59                                               ;ERF GATE ON STOP.
60 01662 020064          LDA     0,IBUFF
```

```
 0025 MT110
01 01663 024114         LDA     1,C10
02 01664 107000         ADD     0,1
03 01665 061422         DIB     0,.MTA
04 01666 030126         LDA     2,C21           ; SET = 21 FOR ERROR PRINT
05 01667 106415         SUB#    0,1,SNR         ;ADDRESS REGISTER WRONG
06 01670 000402         JMP     .+3             ;AC1=CORRECT ADDRESS, AC0=BAD.
07 01671 006314         HALT1                   ;CHECK INCREMENT WC
08 01672 000404         JMP     A35E
09 01673 006303         CHECK
10 01674 000010         10
11 01675 006315         HALT2                   ;DATA ERROR.
12                                              ;AC0=GOOD WORD,  AC1=BAD WORD
13 01676 006277 A35E:   JSR     @LOOP.          ;AC2=ADDRESS OF THE INPUT BUFFER
14                      ;READ 63 CHAR REC. WITH WC=10
15
16 01677 006275 A36:    JSR     @SETP1          ;REWIND-WRITE 1 REC.
17 01700 006317         REWIND                  ;CHECK STATUS-REWIND
18 01701 006270         GEN                     ;READ-CHECK DONE,
19 01702 125252         125252                  ;STATUS,ADDR.  REG.
20 01703 006267         WRITE                   ;AND DATA
21 01704 000177         177
22 01705 006265         STATUS
23 01706 000001         1
24 01707 006300         EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
25 01710 000431         JMP     A36E            ;THIS MUCH HAS WORKED BEFORE
26 01711 006317         REWIND
27 01712 006301         READ
28 01713 000110         110
29 01714 030112         LDA     2,C6            ; SET = 6 FOR ERROR PRINT
30 01715 063722         SKPDZ   .MTA
31 01716 000403         JMP     .+3
32 01717 006314         HALT1                   ;NO DONE FLAG.  CHECK INCREMENT
33 01720 000421         JMP     A36E            ;WC-WC OVERFLOW-STOP
34 01721 006265         STATUS
35 01722 000001         1                       ;AC1=EXPECTED STATUS, AC0=BAD
36 01723 006300         EHALT                   ;THIS IS THE FIRST TIME A READ IS
37 01724 000415         JMP     A36E            ;STOPPED BY WC OVERFLOW ALONE.
38                                              ;LOOK FOR WC OVERFLOW AND CHECK
39                                              ;THE WC OVERFLOW GATE ON STOP
40 01725 020064         LDA     0,IBUFF
41 01726 024114         LDA     1,C10
42 01727 107000         ADD     0,1
43 01730 061422         DIB     0,.MTA
44 01731 030126         LDA     2,C21           ; SET = 21 FOR ERROR PRINT
45 01732 106415         SUB#    0,1,SNR         ;ADDRESS REG. WRONG.  AC1=CORRECT
46 01733 000403         JMP     .+3             ;ADDR.  AC0=BAD.
47 01734 006314         HALT1                   ;CHECK DCHI LOGIC AND INCREMENT WC
48 01735 000404         JMP     A36E            ;DATA ERROR.  INPUT BUFFER SHOULD
49 01736 006303         CHECK                   ;=OUTPUT BUFFER FOR THE FIRST 8
50 01737 000010         10                      ;WORDS.
51 01740 006315         HALT2                   ;AC0=GOOD WORD,  AC1=BAD WORD
52 01741 006277 A36E:   JSR     @LOOP.          ;AC2=ADDRESS OF THE INPUT BUFFER
53 01742 006275 A39:    JSR     @SETP1          ;REWIND-WRITE A 16
54 01743 006317         REWIND                  ;CHAR REC., WRITE EOF
55 01744 006267         WRITE                   ;CHECK STATUS
56 01745 000110         110                     ;REWIND-READ 2 REC.
57 01746 006271         WEOF                    ;CHECK STATUS,CA
58 01747 006265         STATUS
59 01750 100401         100401
60 01751 006300         EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
```

```
 0026 MT110
01 01752 000420          JMP     A39E            ;THIS HAS WORKED BEFORE.
02 01753 006217          REWIND
03 01754 006201          READ
04 01755 000210          210
05 01756 034152          LDA     3,C1004
06 01757 054402          STA     3,.+2
07 01760 006265          STATUS
08 01761 100401          100401
09 01762 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
10 01763 000407          JMP     A39E            ;LOOKING FOR READ EOF.
11 01764 024064          LDA     1,IBUFF
12 01765 125400          INC     1,1
13 01766 061422          DIB     0,.MTA
14 01767 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
15 01770 106414          SUB#    0,1,SZR         ;ADDRESS REG WRONG.  AC1=CORRECT
16 01771 006314          HALT1                   ;AC0=BAD.   EOF SHOULD INCR
17 01772 006277 A39E:    JSR     @LOOP.          ;WC AND ADDR. BY 1
18
19                       ;CHECK READ EOF ON SPACE BACK
20
21 01773 006275 A40:     JSR     @SETP1          ;REWIND-WRITE 2 REC
22 01774 006217          REWIND                  ;WRITE EOF THEN 3 REC
23 01775 006267          WRITE                   ;BACKSPACE 10
24 01776 000210          210                     ;CHECK STATUS AND
25 01777 006271          WEOF                    ;ADDR REG.
26 02000 006265          STATUS
27 02001 100401          100401
28 02002 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
29 02003 000416          JMP     A40E            ;THIS HAS WORKED BEFORE.
30 02004 006267          WRITE
31 02005 000310          310
32 02006 006272          BSPACE
33 02007 000010          10
34 02010 006265          STATUS
35 02011 100401          100401
36 02012 006300          EHALT                   ;AC1=EXPECTED STATUS, AC0=BAD
37 02013 000406          JMP     A40E            ;LOOKING FOR EOF.   CHECK
38                                               ;EOF GATE ON STOP
39 02014 024142          LDA     1,C104
40 02015 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
41 02016 061422          DIB     0,.MTA          ;ADDRESS REG.  IS WRONG.
42 02017 106414          SUB#    0,1,SZR         ;AC1=CORRECT VALUE, AC0=BAD
43 02020 006314          HALT1                   ;CHECK "COUNT WC" AND INCREMENT
44                                               ;SHOULD GET 4 COUNT WC
45 02021 006277 A40E:    JSR     @LOOP.          ;BEFORE STOP
46                       ;CHECK READ EOF ON SPACE FORWARD
47
48 02022 006275 A41:     JSR     @SETP1          ;REWIND-WRITE 2 REC
49 02023 006217          REWIND                  ;WEOF-WRITE 3 REC
50 02024 006267          WRITE                   ;REWIND-SPACE 5
51 02025 000210          210                     ;CHECK STATUS,ADDR
52 02026 006271          WEOF                    ;REG.
53 02027 006265          STATUS
54 02030 100401          100401                  ;AC1=EXPECTED STATUS, AC0=BAD
55 02031 006300          EHALT                   ;THIS HAS WORKED BEFORE.
56 02032 000417          JMP     A41E
57 02033 006267          WRITE
58 02034 000310          310
59 02035 006217          REWIND
60 02036 006305          SPACE
```

```
 0027 MT110
01 02037 000005          5
02 02040 006265          STATUS                 ;AC1=EXPECTED STATUS, AC0=BAD
03 02041 100401          100401                 ;LOOKING FOR EOF STATUS.  THIS IS
04 02042 006300          EHALT                  ;THE SAME AS THE LAST EXCEPT THAT
05 02043 000406          JMP     A41E           ;SPACING IS FORWARD
06 02044 024141          LDA     1,C103
07 02045 030126          LDA     2,C21          ; SET = 21 FOR ERROR PRINT
08 02046 061422          DIB     0,.MTA         ;ADDRESS ERROR.  AC1=CORRECT
09 02047 106414          SUB#    0,1,SZR        ;AC0=BAD.  TWO INCREMENT WC
10 02050 006314          HALT1                  ;SHOULD HAVE OCCURRED
11 02051 006277 A41E:    JSR     @LOOP.         ;EOF GIVES STOP.
```

```
!0020 MT110
01                      ; SPACING CHECK
02                      ; DURING SPACING MANY I/O COMMANDS ARE
03                      ; ISSUED TO GENERATE NOISE
04
05 02052 006275 A50:    JSR     @SETP1          ; WRITE 500 2WD REC.
06 02053 006317         REWIND                  ; SPACEBACK 500
07 02054 060422         DIA     0,.MTA          ; CHECK CA
08 02055 101203         MOVR    0,0,SNC
09 02056 000776         JMP     .-2
10 02057 020160         LDA     0,D500
11 02060 040244         STA     0,RECTR
12 02061 006267 A50.1:  WRITE
13 02062 000102         102
14 02063 014244         DSZ     RECTR
15 02064 000775         JMP     A50.1
16 02065 006320         BSPNS
17 02066 000764         500.
18 02067 064422 A50.2:  DIA     1,.MTA
19 02070 064422         DIA     1,.MTA
20 02071 064422         DIA     1,.MTA
21 02072 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
22 02073 125203         MOVR    1,1,SNC
23 02074 000773         JMP     A50.2
24 02075 061422         DIB     0,.MTA
25 02076 024104         LDA     1,CRCNT
26 02077 030126         LDA     2,C21           ; SET = 21 FOR ERROR PRINT
27 02100 122415         SUB#    1,0,SNR         ;ADDRESS REG. WRONG.
28 02101 000403         JMP     .+3             ;SHOULD HAVE COUNTED 500
29 02102 006314         HALT1                   ;REC ON BACKSPACE.
30 02103 000407         JMP     A50E            ;AC1=EXPECTED COUNT, AC0=ACTUAL
31 02104 006272         BSPACE
32 02105 000001         1
33 02106 064422         DIA     1,.MTA
34 02107 020145         LDA     0,C200          ;BACKSPACE 1 DID NOT PRODUCE
35 02110 107415         AND#    0,1,SNR         ;BOT.  ADDR REG COUNTED 500.
36 02111 006300         EHALT                   ;WC REG WAS COUNTING FASTER
37 02112 006277 A50E:   JSR     @LOOP.          ;BAD WC REG.
38                      ;SPACING CHECK.
39                      ;DURING SPACING MANY IO COMMANDS ARE
40                      ;ISSUED TO GENERATE NOISE.
41
42 02113 006275 A51:    JSR     @SETP1          ;WRITE 500 2WD REC.
43 02114 006317         REWIND                  ;SPACEBACK 500
44 02115 060422         DIA     0,.MTA          ;CHECK CA
45 02116 101203         MOVR    0,0,SNC         ;SPACEBACK 1 AND
46 02117 000776         JMP     .-2             ;CHECK FOR BOT
47 02120 020160         LDA     0,D500
48 02121 040244         STA     0,RECTR
49 02122 006267 A51.1:  WRITE
50 02123 000102         102
51 02124 014244         DSZ     RECTR
52 02125 000775         JMP     A51.1
53 02126 006320         BSPNS                   ;BACKSPACE-NO STALL
54 02127 000764         500.                    ; REC COUNT
55 02130 065422 A51.2:  DIB     1,.MTA
56 02131 065422         DIB     1,.MTA
57 02132 064422         DIA     1,.MTA
58 02133 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
59 02134 125203         MOVR    1,1,SNC
60 02135 000773         JMP     A51.2
```

```
 0029 MT110
01 02136 061422          DIB     0,.MTA
02 02137 024104          LDA     1,CRCNT
03 02140 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
04 02141 122415          SUB#    1,0,SNR         ;ADDRESS REG.  WRONG.
05 02142 000403          JMP     .+3             ;SHOULD HAVE COUNTED 500
06 02143 006314          HALT1                   ;REC ON BACKSPACE.
07 02144 000407          JMP     A51E            ;AC1=EXPECTED COUNT, AC0=ACTUAL
08 02145 006272          BSPACE
09 02146 000001          1
10 02147 064422          DIA     1,.MTA
11 02150 020145          LDA     0,C200          ;BACKSPACE 1 DID NOT PRODUCE
12 02151 107415          AND#    0,1,SNR         ;BOT.   ADDR REG COUNTED 500.
13 02152 006300          EHALT                   ;WC REG WAS COUNTING FASTER
14 02153 006277 A51E:    JSR     @LOOP.          ;BAD WC REG.
15                       ;SPACING CHECK.
16                       ;DURING SPACING MANY IO COMMANDS ARE
17                       ;ISSUED TO GENERATE NOISE.
18
19 02154 006275 A52:     JSR     @SETP1          ;WRITE 500 2WD REC.
20 02155 006317          REWIND                  ;SPACEBACK 500
21 02156 060422          DIA     0,.MTA          ;CHECK CA
22 02157 101202          MOVR    0,0,SNC         ;SPACEBACK 1 AND
23 02160 000776          JMP     .-2             ;CHECK FOR BOT
24 02161 020160          LDA     0,D500
25 02162 040244          STA     0,RECTR
26 02163 006267 A52.1:   WRITE
27 02164 000102          102
28 02165 014244          DSZ     RECTR
29 02166 000775          JMP     A52.1
30 02167 006320          BSPNS                   ;BACKSPACE-NO STALL
31 02170 000764          500                     ; REC COUNT
32 02171 065422 A52.2:   DIB     1,.MTA
33 02172 065422          DIB     1,.MTA
34 02173 064422          DIA     1,.MTA
35 02174 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
36 02175 125203          MOVR    1,1,SNC
37 02176 000773          JMP     A52.2
38 02177 061422          DIB     0,.MTA
39 02200 024104          LDA     1,CRCNT
40 02201 030126          LDA     2,C21           ; SET = 21 FOR ERROR PRINT
41 02202 122415          SUB#    1,0,SNR         ;ADDRESS REG.  WRONG.
42 02203 000403          JMP     .+3             ;SHOULD HAVE COUNTED 500
43 02204 006314          HALT1                   ;REC ON BACKSPACE.
44 02205 000407          JMP     A52E            ;AC1=EXPECTED COUNT, AC0=ACTUAL
45 02206 006272          BSPACE
46 02207 000001          1
47 02210 064422          DIA     1,.MTA
48 02211 020145          LDA     0,C200          ;BACKSPACE 1 DID NOT PRODUCE
49 02212 107415          AND#    0,1,SNR         ;BOT.   ADDR REG COUNTED 500.
50 02213 006300          EHALT                   ;WC REG WAS COUNTING FASTER
51 02214 006277 A52E:    JSR     @LOOP.          ;BAD WC REG.
52                       ;SPACING CHECK.
53                       ;DURING SPACING MANY IO COMMANDS ARE
54                       ;ISSUED TO GENERATE NOISE.
55
56 02215 006275 A53:     JSR     @SETP1          ;WRITE 500 2WD REC.
57 02216 006317          REWIND                  ;SPACEBACK 500
58 02217 060422          DIA     0,.MTA          ;CHECK CA
59 02220 101202          MOVR    0,0,SNC         ;SPACEBACK 1 AND
60 02221 000776          JMP     .-2             ;CHECK FOR BOT
```

```
 0030 MT110
01 02222 020160         LDA     0,D500
02 02223 040244         STA     0,RECTR
03 02224 006267 A53.1:  WRITE
04 02225 000102         102
05 02226 014244         DSZ     RECTR
06 02227 000775         JMP     A53.1
07 02230 006320         BSPNS                   ;BACKSPACE-NO STALL
08 02231 000764         500.                    ; REC COUNT
09 02232 065422 A53.2:  DIB     1,.MTA
10 02233 065422         DIB     1,.MTA
11 02234 054422         DIA     1,.MTA
12 02235 125203         MOVR    1,1,SNC
13 02236 000774         JMP     A53.2
14 02237 061422         DIB     0,.MTA
15 02240 024104         LDA     1,CRONT
16 02241 030126         LDA     2,C21           ; SET = 21 FOR ERROR PRINT
17 02242 122415         SUB#    1,0,SNR         ;ADDRESS REG.  WRONG.
18 02243 000403         JMP     .+3             ;SHOULD HAVE COUNTED 500
19 02244 006314         HALT1                   ;REC ON BACKSPACE.
20 02245 000407         JMP     A53E            ;AC1=EXPECTED COUNT, AC0=ACTUAL
21 02246 006272         BSPACE
22 02247 000001         1
23 02250 064422         DIA     1,.MTA
24 02251 020145         LDA     0,C200          ;BACKSPACE 1 DID NOT PRODUCE
25 02252 107415         AND#    0,1,SNR         ;BOT.   ADDR REG COUNTED 500.
26 02253 006300         EHALT                   ;WC REG WAS COUNTING FASTER
27 02254 006277 A53E:   JSR     @LOOP.          ;BAD WC REG.
28                      ;LONGITUDINAL PARITY CHECK
29
30 02255 020455         LDA     0,LSTRT         ;CAUSE LONG PARITY ERROR
31 02256 040453         STA     0,LPOIT         ;IN 2 TRK PAIRS
32 02257 102400         SUB     0,0             ;CHECK PARITY STATUS
33 02260 040450         STA     0,RBCT
34 02261 006266         ERASE
35 02262 006270 A54:    GEN
36 02263 037477         37477
37 02264 020063         LDA     0,OBUFF
38 02265 101400         INC     0,0
39 02266 040170         STA     0,IIDX0
40 02267 022442         LDA     0,@LPOIT
41 02270 101005         MOV     0,0,SNR
42 02271 000442         JMP     CKED
43 02272 010170 A54.1:  ISZ     IIDX0           ;%
44 02273 042170         STA     0,@IIDX0
45 02274 102400         SUB     0,0
46 02275 024165         LDA     1,M400
47 02276 010170         ISZ     IIDX0           ;%
48 02277 042170         STA     0,@IIDX0
49 02300 125404         INC     1,1,SZR
50 02301 000775         JMP     .-3
51 02302 006275         JSR     @SETP1
52 02303 020063         LDA     0,OBUFF         ; GET OUTPUT BUFFER ADDRESS
53 02304 062022         DOB     0,.MTA          ; SET MA REGISTER
54 02305 020165         LDA     0,M400          ; GET RECORD LENGTH
55 02306 063222         DOCC    0,.MTA          ; SET WORD COUNT AND CLEAR
56 02307 020144         LDA     0,C150          ; GET EVEN PARITYU WRITE
57 02310 024234         LDA     1,CX            ; GET UNIT NUMBER
58 02311 122000         ADD     1,0             ; CREATE WRITE COMMAND
59 02312 061122         DOAS    0,.MTA          ; START THE WRITE
60 02313 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
```

```
 0031 MT110
01 02314 063522          SKPBZ   .MTA            ; TEST FOR WRITE DONE
02 02315 000776          JMP     .-2             ; IF NOT DONE, TEST AGAIN
03 02316 006267          WRITE                   ;NO PARITY ERROR.  # IN AC2
04 02317 000102          102                     ;INDICATES WHICH TRACKS SHOULD
05 02320 125000          MOV     1,1             ;CAUSE ERROR. (SEE BELOW)
06 02321 125000          MOV     1,1             ;0=RB2-P 6=RB2-3  14=RB0-P
07 02322 125000          MOV     1,1             ;1=RB3-P 7=RB3-4  15=RB1-P
08                                               ;2=RB4-P 10=RB4-5 16=RB0-1
09                                               ;3=RB5-P 11=RB5-6 17=RB0-2
10 02323 000401          JMP     .+1             ;4=RB6-P 12=RB6-7 20=RB1-2
11 02324 006277          JSR     @LOOP.          ;5=RB7-P 13=RB7-2
12 02325 010404          ISZ     LPOIT
13 02326 010402          ISZ     RBCT
14 02327 000733          JMP     A54
15
16 02330 000000 RBCT:    0
17 02331 000000 LPOIT:   0
18 02332 004533 LSTRT:   LST79
19 02333 060422 CKED:    DIA     0,.MTA
20 02334 024136          LDA     1,C100
21 02335 010774          ISZ     LPOIT
22 02336 123405          AND     1,0,SNR
23 02337 000404          JMP     .+4
24 02340 022771          LDA     0,@LPOIT
25 02341 101004          MOV     0,0,SZR
26 02342 000730          JMP     A54.1
27 02343 006317          REWIND                  ;TEST END
28 02344 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
29 02345 006265          STATUS
30 02346 100201          100201
31 02347 000775          JMP     .-3
32 02350 000401          JMP     .+1
33
34                       ;CHECK FOR DELETE WRITE LOCK
35
36 02351 020203 ETST:    LDA     0,PASS          ; GET PASS COUNTER
37 02352 101004          MOV     0,0,SZR         ; TEST FOR FIRST PASS
38 02353 000441          JMP     ET.T4           ; IF NOT FIRST, SKIP TESTS
39 02354 022205          LDA     0,@ISWR?EG
40 02355 126520          SUBZL   1,1             ; SET MASK BIT
41 02356 123414          AND#    1,0,SZR
42 02357 000420          JMP     ET.T3
43 02360 006307          JSR     @PINWL
44 02361 006317          REWIND
45 02362 006265          STATUS
46 02363 100205          100205
47 02364 000404          JMP     .+4             ;RING IN
48 02365 000401          JMP     .+1
49 02366 006207          MESSAGE
50 02367 004510          PRB
51 02370 060222          NIOC    .MTA            ; CLEAR CONTROLLER
52 02371 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
53 02372 006317          REWIND
54 02373 006265          STATUS
55 02374 100201          100201
56 02375 000773          JMP     .-5
57 02376 000401          JMP     .+1
58
59                       ;CHECK FOR DELETE EOT TEST
60
```

```
 0032 MT110
01 02377 032205 ET.T3:  LDA     2,@ISWR?EG
02 02400 034105         LDA     3,C2            ; GET MASK BIT
03 02401 173414         AND#    3,2,SZR
04 02402 000402         JMP     .+2
05 02403 006310         JSR     @PETTA
06 02404 006207         MESSAGE
07 02405 004422         MCYC
08 02406 006317         REWIND
09 02407 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
10 02410 006265         STATUS
11 02411 100201         100201
12 02412 000775         JMP     .-3
13 02413 000401         JMP     .+1
14 02414 010203 ET.T4:  ISZ     PAS?S           ; COUNT THE PASS
15 02415 101001         MOV     0,0,SKP         ; SKIP IF COUNTER NOT FULL
16 02416 010203         ISZ     PAS?S           ; SET TO ONE IF FULL
17 02417 022205         LDA     0,@ISWR?EG      ; GET SWREG VALUE
18 02420 024153         LDA     1,C4K           ; GET PASS PRINT BIT
19 02421 107414         AND#    0,1,SZR         ; TEST FOR PRINT PASS
20 02422 002406         JMP     @ET.T5          ; IF SET, NO PRINT
21 02423 006207         MESSAGE
22 02424 003420         MSG2
23 02425 024203         LDA     1,PAS?S         ; GET THE PASS NUMBER
24 02426 006212         TYPED                   ; PRINT IT IN DECIMAL
25 02427 002401         JMP     @.+1
26
27 02430 000531 ET.T5:  A1
28
```

```
!0033 MT110
01                      ;MTA DIAGNOSTIC INITIALIZATION
02
03 02431 054240 INITIAL: STA    3,CNTR          ; SAVE RETURN ADDRESS
04 02432 102400         SUB     0,0             ; CLEAR REGISTER
05 02433 040203         STA     0,PASS          ; INITIALIZE PASS COUNTER
06 02434 006207         MESSAGE                 ;NAME
07 02435 006765         DIRT
08 02436 006207         MESSAGE
09 02437 003360         MSG1                    ;PRINT HALT, MOUNT SCRATCH PACK
10 02440 006222         ASCIN                   ; GETT CONTINUE KEY
11 02441 006211         TYPEA                   ; PRINT THE CHARACTER
12 02442 006207 INIT1:  MESSAGE                 ; PRINT UNIT REQUEST
13 02443 003424         MSGUN
14 02444 006216         OCTIN                   ; GET UNIT NUMBER
15 02445 000775         JMP     INIT1           ; IF ERROR REQUEST AGAIN
16 02446 020114         LDA     0,C10           ; GET MAX NUMBER
17 02447 106432         SUBZ#   0,1,SZC         ; TEST BETWEEN 0 AND 7
18 02450 000772         JMP     INIT1           ; IF NOT, REQUEST AGAIN
19 02451 135000         MOV     1,3             ; SAVE UNIT NUMBER
20 02452 030114         LDA     2,C10           ; GET COMMAND
21 02453 173000         ADD     3,2             ;AC2=UNIT # +10
22 02454 050233         STA     2,C1X           ;STORE UNIT#+10
23 02455 054234         STA     3,CX            ;SAVE UNIT#
24 02456 034131         LDA     3,C40           ; GET COMMAND
25 02457 173000         ADD     3,2             ; AC2 = UNIT # + 50
26 02460 050235         STA     2,C5X           ; STORE UNIT # + 50
27 02461 050236         STA     2,CC5X
28 02462 034124         LDA     3,C20           ; GET COMMAND
29 02463 173000         ADD     3,2             ; SET COMMAND + UNIT
30 02464 050237         STA     2,C7X           ; SAVE ERASE COMMAND
31 02465 006207 INIT6:  MESSAGE                 ; ENTER DEVICE CODE 22 OR 62
32 02466 003435         MSGDC
33 02467 006216         OCTIN                   ; GET OPERATOR RESPONSE
34 02470 000775         JMP     INIT6           ; IF ERROR, REQUEST AGAIN
35 02471 020127         LDA     0,C22           ; GET NORMAL DEVICE CODE
36 02472 106415         SUB#    0,1,SNR         ; TEST FOR NORMAL
37 02473 000404         JMP     INIT7           ; IF NORMAL SAVE IT
38 02474 020133         LDA     0,C62           ; GET SECONDARY CODE
39 02475 106414         SUB#    0,1,SZR         ; TEST FOR SECONDARY
40 02476 000767         JMP     INIT6           ; IF NOT, REQUEST AGAIN
41 02477 040243 INIT7:  STA     0,DEVICE        ; SAVE THE NEW CODE
42
43                      ;MODIFY ALL .MTA INSTRUCTIONS
44
45 02500 030060         LDA     2,FIRST
46 02501 021000 GMOR:   LDA     0,0,2
47 02502 024075         LDA     1,C160077
48 02503 123400         AND     1,0
49 02504 026076         LDA     1,@PCDCM
50 02505 106414         SUB#    0,1,SZR
51 02506 000407         JMP     DEV1            ;NOT MTA
52 02507 021000         LDA     0,0,2
53 02510 024077         LDA     1,C177700
54 02511 123400         AND     1,0
55 02512 024243         LDA     1,DEVICE
56 02513 123000         ADD     1,0
57 02514 041000         STA     0,0,2
58 02515 151400 DEV1:   INC     2,2
59 02516 020061         LDA     0,CEND
60 02517 112414         SUB#    0,2,SZR
```

```
 0034 MT110
02520 000761         JMP     GMOR
02521 006223 INIT8:  CALIB                   ; CALIBRATE THE TIME BASE
02522 060214         NIOC    RTC             ; CLEAR THE REAL TIME CLOCK
02523 006207         MESSAGE                 ; PRINT SWREG SET MESSAGE
02524 003455         MSGS
02525 034221 INT10:  LDA     3,IOM?OD        ; GET I/O MOD FLAG
02526 175005         MOV     3,3,SNR         ; TEST FOR MOD SET
02527 000404         JMP     INT11           ; IF NOT SET, TTY INPUT
02530 006221         JSR     @IOM?OD         ; INPUT TO AC0
02531 000001         1                       ; INPUT CONTROL FLAG
02532 000403         JMP     INT12           ; GO TEST SWREG ENTRY
02533 063610 INT11:  SKPDN   TTI             ; TEST FOR TTY READY
02534 000777         JMP     .-1             ; IF NOT READY, WAIT
02535 006206 INT12:  JSR     @IINP?          ; TEST FOR SWITCH COMMAND
02536 020123         LDA     0,CONO          ; GET ASCII CONTROL CHARACTER
02537 116415         SUB#    0,3,SNR         ; TEST FOR ODT REQUEST
02540 000405         JMP     INT13           ; IF ODT, GO ENTER IT
02541 020121         LDA     0,ASCR          ; GET ASCII CR CHARACTER
02542 116415         SUB#    0,3,SNR         ; TEST FOR EXIT REQUEST
02543 000403         JMP     INIT9           ; IF CR, EXIT SWREG ENTRY
02544 000761         JMP     INT10           ; IF NOT CR, GET NEXT ENTRY
02545 006226 INT13:  JSR     @IODT?          ; ENTER ODT HERE
02546 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

02547 054444 XCHK:   STA     3,RCHK
02550 024063         LDA     1,OBUFF
02551 044172         STA     1,IIDX2
02552 014172         DSZ     IIDX2
02553 024064         LDA     1,IBUFF
02554 044173         STA     1,IIDX3
02555 014173         DSZ     IIDX3
02556 031400         LDA     2,0,3           ;GET ARG
02557 050240         STA     2,CNTR
02560 010172 XCHK.:  ISZ     IIDX2           ;%
02561 022172         LDA     0,@IIDX2
02562 010173         ISZ     IIDX3           ;%
02563 026173         LDA     1,@IIDX3
02564 106414         SUB#    0,1,SZR
02565 000407         JMP     LEAV
02566 014240         DSZ     CNTR
02567 000771         JMP     XCHK.
02570 010173         ISZ     IIDX3           ;%
02571 026173         LDA     1,@IIDX3
02572 125005         MOV     1,1,SNR         ;WORD FOLLOWING LAST
                                             ;MUST BE ZERO
02573 010420         ISZ     RCHK
02574 010417 LEAV:   ISZ     RCHK
```

```
 0035 MT110
01 02575 030064          LDA     2,IBUFF
02 02576 002415          JMP     @RCHK
03
04                       ;CLEAR INPUT BUFFER TO ZEROS
05
06 02577 020064 XCLR:    LDA     0,IBUFF
07 02600 126000          ADC     1,1
08 02601 123000          ADD     1,0
09 02602 040172          STA     0,IIDX2
10 02603 020136          LDA     0,C100
11 02604 040240          STA     0,CNTR
12 02605 102400          SUB     0,0
13 02606 010172          ISZ     IIDX2           ;%
14 02607 042172          STA     0,@IIDX2
15 02610 014240          DSZ     CNTR
16 02611 000775          JMP     .-3
17 02612 001400          JMP     0,3
18
19 02613 000000 RCHK:    0
20
21                       ;DO A SPACE BACKWARD
22
23 02614 152401 .XBSP:   SUB     2,2,SKP
24 02615 152520 XBSP:    SUBZL   2,2
25 02616 024131          LDA     1,C40
26 02617 000403          JMP     XSPC0
27
28                       ;DO A SPACE FORWARD
29
30 02620 152520 XSPC:    SUBZL   2,2
31 02621 024130          LDA     1,C30
32 02622 050433 XSPC0:   STA     2,WSW
33 02623 044431          STA     1,CINS
34 02624 025400          LDA     1,0,3
35 02625 124400          NEG     1,1
36 02626 057022          DOC     1,.MTA          ;SET REC COUNT
37 02627 175400          INC     3,3
38 02630 054423          STA     3,XSPRET        ;RETURN ADDR
39 02631 024136          LDA     1,C100
40 02632 066022          DOB     1,.MTA          ;CA REG = 100
41 02633 024421          LDA     1,CINS
42 02634 030234          LDA     2,CX
43 02635 147000          ADD     2,1
44 02636 065122          DOAS    1,.MTA          ;SPACE !!
45 02637 014416          DSZ     WSW
46 02640 002413          JMP     @XSPRET         ;DONT WAIT FOR DONE
47 02641 024125          LDA     1,C20.
48 02642 044245          STA     1,WCTR
49 02643 006224 XSPC1:   WAIT
50 02644 000067          .200MS
51 02645 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
52 02646 063722          SKPDZ   .MTA
53 02647 002404          JMP     @XSPRET         ;DONE
54 02650 014245          DSZ     WCTR
55 02651 000772          JMP     XSPC1
56 02652 002401          JMP     @XSPRET         ;TIMEOUT, 1 SEC
57 02653 000000 XSPRET:  0
58 02654 000000 CINS:    0
59 02655 000000 WSW:     0
60
```

```
 0036 MT110
01                      ;REWIND SUBROUTINE, NO WAIT
02
03 02656 024233 .RWNS:  LDA     1,C1X
04 02657 065122         DOAS    1,.MTA
05 02660 001400         JMP     0,3
06
07                      ;REWIND AND WAIT 20 SEC
08                      ;OR UNTIL "DONE" AND "TUR"
09
10 02661 054440 .RW:    STA     3,.RWRET
11 02662 020234         LDA     0,CX
12 02663 061022         DOA     0,.MTA          ;SELECT UNIT
13 02664 020103         LDA     0,C100.
14 02665 040245         STA     0,WCTR
15 02666 060422 RW0:    DIA     0,.MTA          ;READ STATUS
16 02667 034151         LDA     3,C20K
17 02670 163414         AND#    3,0,SZR
18 02671 000413         JMP     .RW2-2          ;REWINDING NOW
19 02672 101202         MOVR    0,0,SZC         ;WAIT FOR "TUR"
20 02673 000407         JMP     RW1             ;TUR=1
21 02674 006224         WAIT
22 02675 000067         .200MS
23 02676 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
24 02677 014245         DSZ     WCTR
25 02700 000766         JMP     RW0
26 02701 002420         JMP     @.RWRET         ;20 SEC. TIMEOUT, NO TUR
27
28 02702 024233 RW1:    LDA     1,C1X
29 02703 065122         DOAS    1,.MTA          ;REWIND !!
30 02704 024103         LDA     1,C100.
31 02705 044245         STA     1,WCTR
32 02706 006224 .RW2:   WAIT
33 02707 000067         .200MS
34 02710 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
35 02711 064422         DIA     1,.MTA          ;READ STATUS
36 02712 020146         LDA     0,C201          ;TUR + BOT
37 02713 107400         AND     0,1
38 02714 106405         SUB     0,1,SNR
39 02715 002404         JMP     @.RWRET
40 02716 014245         DSZ     WCTR
41 02717 000767         JMP     .RW2
42 02720 002401         JMP     @.RWRET         ;TIMEOUT 20 SEC
43
44 02721 000000 .RWRET: 0
45
46
47                      ;WRITE EOF
48                      ;WAIT 1 SEC OR UNTIL DONE
49
50 02722 054420 XWEOF:  STA     3,RWEOF
51 02723 024234         LDA     1,CX
52 02724 030132         LDA     2,C60
53 02725 147000         ADD     2,1
54 02726 065122         DOAS    1,.MTA
55 02727 020125         LDA     0,C20.
56 02730 040245         STA     0,WCTR
57 02731 006224         WAIT
58 02732 000066         .50MS
59 02733 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
60 02734 063722         SKPDZ   .MTA
```

```
 0037 MT110
01 02735 002405          JMP     @RWEOF
02 02736 014245          DSZ     WCTR
03 02737 000772          JMP     .-6
04 02740 002402          JMP     @RWEOF
05 02741 002401          JMP     @RWEOF
06
07 02742 000000 RWEOF:   0
08                       ;LOAD OBUFF WITH 3 WORDS.
09                       ;
10                       ;       LOAD
11                       ;       ARG
12                       ;
13                       ;ARGUMENT=ADDR OF 3 WORD FIELD
14
15 02743 031400 XLD:     LDA     2,0,3
16 02744 050173          STA     2,IIDX3
17 02745 030063          LDA     2,OBUFF
18 02746 050174          STA     2,IIDX4
19 02747 014174          DSZ     IIDX4
20 02750 102000          ADC     0,0
21 02751 024161          LDA     1,M3
22 02752 010173          ISZ     IIDX3           ;%
23 02753 032173          LDA     2,@IIDX3
24 02754 113400          AND     0,2
25 02755 010174          ISZ     IIDX4           ;%
26 02756 052174          STA     2,@IIDX4
27 02757 125404          INC     1,1,SZR
28 02760 000772          JMP     .-6
29 02761 175400          INC     3,3
30 02762 001400          JMP     0,3
31
32                       ;GENERATE ARGUMENT DATA INTO OBUFF
33                       ;ALWAYS GENERATE 64 WORDS ALL SAME
34                       ;
35                       ;       GEN
36                       ;       ARG
37                       ;
38
39 02763 054416 XGEN:    STA     3,RGEN
40 02764 010415          ISZ     RGEN
41 02765 031400          LDA     2,0,3           ;GET DATA WORD
42 02766 102000          ADC     0,0
43 02767 024063          LDA     1,OBUFF
44 02770 044170          STA     1,IIDX0         ;SET IR0
45 02771 014170          DSZ     IIDX0
46 02772 024164          LDA     1,M100          ;LOAD 64 COUNTER
47 02773 044240          STA     1,CNTR
48 02774 010170          ISZ     IIDX0           ;%
49 02775 052170          STA     2,@IIDX0
50 02776 010240          ISZ     CNTR
51 02777 000775          JMP     .-3
52 03000 002401          JMP     @RGEN
53
54 03001 000000 RGEN:    0
55
56                       ;SELECT THE UNIT
57
58 03002 054412 XSEL:    STA     3,RSEL
59 03003 006224          WAIT
60 03004 000065          .25MS
```

```
 0038 MT110
01 03005 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
02 03006 020237          LDA     0,C7X
03 03007 061022          DOA     0,.MTA
04 03010 006224          WAIT
05 03011 000065          .25MS
06 03012 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
07 03013 002401          JMP     @RSEL
08
09 03014 000000 RSEL:    0
10                       ;READ SUBROUTINE
11                       ;ARGUMENT FOLLOWING CALL SPECIFIES
12                       ;#WORDS, AND PARITY TYPE
13                       ;BITS 10-15=WORDS       BIT 0=  1 FOR EVEN
14                       ;BITS 4-9=RECORDS               0 FOR ODD
15                       ;                       BIT 1=  1 FOR TEST MODE
16
17 03015 054450 XRD:     STA     3,RRD
18 03016 010447          ISZ     RRD
19 03017 031400          LDA     2,0,3
20 03020 020135          LDA     0,C77
21 03021 143400          AND     2,0
22 03022 100400          NEG     0,0
23 03023 040241          STA     0,WDCNT         ;SAVE WORD COUNT
24 03024 102400          SUB     0,0
25 03025 151112          MOVL#   2,2,SZC         ;SKIP IF ODD PAR
26 03026 020136          LDA     0,C100
27 03027 024234          LDA     1,CX
28 03030 123000          ADD     1,0             ;AC0=READ INST.
29 03031 040242          STA     0,INST
30 03032 020154          LDA     0,C7700
31 03033 143400          AND     2,0             ;AC0=#RECX100
32 03034 024064          LDA     1,IBUFF
33 03035 125120          MOVZL   1,1
34 03036 153132          ADDZL#  2,2,SZC
35 03037 125241          MOVOR   1,1,SKP
36 03040 125220          MOVZR   1,1
37 03041 044425          STA     1,CASAV         ;SAVE CA
38 03042 024136          LDA     1,C100          ;AC1=100
39 03043 030241 XRD.1:   LDA     2,WDCNT
40 03044 073022          DOC     2,.MTA          ;SET WC
41 03045 030421          LDA     2,CASAV
42 03046 072022          DOB     2,.MTA          ;LOAD CA REG
43 03047 030242          LDA     2,INST
44 03050 071122          DOAS    2,.MTA          ;START READ
45 03051 030125          LDA     2,C20.
46 03052 050245          STA     2,WCTR
47 03053 006224          WAIT
48 03054 000066          .50MS
49 03055 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
50 03056 063722          SKPDZ   .MTA
51 03057 000403          JMP     .+3
52 03060 014245          DSZ     WCTR
53 03061 000772          JMP     .-6
54 03062 122404          SUB     1,0,SZR
55 03063 000760          JMP     XRD.1
56 03064 002401          JMP     @RRD
57
58 03065 000000 RRD:     0
59 03066 000000 CASAV:   0
60                       ;WRITE NO STALL (WRTNS)
```

```
 0039 MT110
01                      ;JUST LIKE (WRITE) EXCEPT THAT ONLY
02                      ;ONE RECORD IS WRITTEN AND RETURN
03                      ;IS EXECUTED IMMEDIATELY
04
05 03067 102000 XWNS:   ADC     0,0
06 03070 040453         STA     0,SWTCH
07 03071 000403         JMP     XWRT+2
08
09                      ;WRITE SUBROUTINE
10                      ;ARGUMENT FOLLOWING CALL SPECIFIES
11                      ;#RECORDS AND WORDS, AND PARITY.
12                      ;BITS10-15=#WORDS       BIT 0 = 1FOR EVEN
13                      ;BITS 4- 9=#RECORDS             0 FOR ODD
14
15 03072 102400 XWRT:   SUB     0,0
16 03073 040450         STA     0,SWTCH
17 03074 054446         STA     3,RWRT
18 03075 010445         ISZ     RWRT
19 03076 031400         LDA     2,0,3
20 03077 020135         LDA     0,C77
21 03100 113400         AND     0,2
22 03101 150400         NEG     2,2
23 03102 050241         STA     2,WDCNT         ;SAVE WORD COUNT
24 03103 031400         LDA     2,0,3
25 03104 102400         SUB     0,0
26 03105 151112         MOVL#   2,2,SZC         ;SKIP IF ODD PAR
27 03106 020136         LDA     0,C100
28 03107 024235         LDA     1,CSX
29 03110 123000         ADD     1,0
30 03111 040242         STA     0,INST          ;SAVE INST
31 03112 020154         LDA     0,C7700
32 03113 143400         AND     2,0             ;AC0=#RECX100
33 03114 024136         LDA     1,C100
34 03115 030241 XWRT.:  LDA     2,WDCNT
35 03116 073222         DOCC    2,.MTA          ;SET WC
36 03117 030063         LDA     2,OBUFF
37 03120 072022         DOB     2,.MTA          ;SET CA
38 03121 030242         LDA     2,INST
39 03122 071122         DOAS    2,.MTA          ;START WRITE
40 03123 030420         LDA     2,SWTCH         ;CHECK SWITCH
41 03124 151004         MOV     2,2,SZR
42 03125 002415         JMP     @RWRT
43 03126 030125         LDA     2,C20.
44 03127 050245         STA     2,WCTR
45 03130 006224         WAIT
46 03131 000066         .50MS
47 03132 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
48 03133 063722         SKPDZ   .MTA
49 03134 000403         JMP     .+3
50 03135 014245         DSZ     WCTR
51 03136 000772         JMP     .-6
52 03137 122404         SUB     1,0,SZR
53 03140 000755         JMP     XWRT.
54 03141 002401         JMP     @RWRT
55
56 03142 000000 RWRT:   0
57 03143 000000 SWTCH:  0
58                      ;ERASE ROUTINE
59
60 03144 020234 XERAS:  LDA     0,CX
```

```
 0040 MT110
01 03145 054417          STA     3,XERAR
02 03146 024134          LDA     1,C70
03 03147 123000          ADD     1,0
04 03150 061122          DOAS    0,.MTA
05 03151 020125          LDA     0,C20.
06 03152 040245          STA     0,WCTR
07 03153 020124          LDA     0,C20
08 03154 063522          SKPBZ   .MTA
09 03155 000777          JMP     .-1
10 03156 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
11 03157 063722          SKPDZ   .MTA
12 03160 002404          JMP     @XERAR
13 03161 014245          DSZ     WCTR
14 03162 000772          JMP     .-6
15 03163 002401          JMP     @XERAR
16 03164 000000 XERAR:   0
17
18                       ;CHECK STATUS FOR TUR AND ERASE.
19                       ;AC0=UNIT BEING SELECTED.
20                       ;RETURN+1 IF ERROR
21
22 03165 054421 CSTAT:   STA     3,CSTAR
23 03166 034234          LDA     3,CX            ; GET SELECTED UNIT NUMBER
24 03167 024113          LDA     1,C7
25 03170 123400          AND     1,0             ;AC0=ACTUAL UNIT BEING SEL.
26 03171 162415          SUB#    3,0,SNR
27 03172 000407          JMP     ON              ;TUR SHOULD BE ON
28 03173 126520          SUBZL   1,1             ;UNIT NOT SEL.
29 03174 060422          DIA     0,.MTA
30 03175 123415          AND#    1,0,SNR         ;CORRECT STATUS = 0
31 03176 010410 OKRET:   ISZ     CSTAR           ;OK, RETURN+2
32 03177 030232          LDA     2,X70           ;X70=UNIT#+70
33 03200 002406          JMP     @CSTAR
34 03201 006265 ON:      STATUS
35 03202 100201          100201
36 03203 000774          JMP     OKRET+1         ;ERROR,AC1=GOOD STAT.
37 03204 101000          MOV     0,0
38 03205 000771          JMP     OKRET           ;AC0=BAD
39
40 03206 000000 CSTAR:   0
41                       ;COMBINE ARGUMENT STATUS WITH SWR
42                       ;STATUS AND CHECK AGAINST ACTUAL.
43                       ;RETURN+3 IF STATUS MATCH, +1 OTHERWISE
44                       ;EXIT WITH AC1=EXPECTED STATUS, AC0=ACTUAL
45                       ;
46                       ;       STATUS
47                       ;       ARG
48                       ;
49
50 03207 054414 XSTAT:   STA     3,STATR
51 03210 010413          ISZ     STATR
52 03211 025400          LDA     1,0,3
53 03212 034136          LDA     3,C100          ;9TRK=1
54 03213 167000          ADD     3,1             ; IF 9 TRACK, SET BIT
55 03214 034153          LDA     3,C4K           ;HI DENS=1
56 03215 167000          ADD     3,1             ;AC1=EXPECTED STATUS
57 03216 060422          DIA     0,.MTA          ;AC0=ACTUAL STATUS
58 03217 034404          LDA     3,STATR
59 03220 106415          SUB#    0,1,SNR
60 03221 001402          JMP     2,3
```

```
 0041 MT110
01 03222 001400          JMP     0,3
02
03 03223 000000 STATR:   0
04                       ;END TAPE TEST.  WRITE TO EOT AND
05                       ;HALT AFTER MESSAGE.  NOTE ERRORS
06                       ;ALONG THE WAY
07
08 03224 006264 ETT:     JSR     @INTIL
09 03225 004406          JSR     ETTA
10 03226 006317          REWIND                  ; REWIND TO BOT AT END
11 03227 006207          MESSAGE
12 03230 003514          MSGTC
13 03231 006226          JSR     @IODT?          ; END OF TEST MESSAGE
14 03232 000777          JMP     .-1
15 03233 054427 ETTA:    STA     3,ETTB
16 03234 006275          JSR     @SETP1          ; SETUP FOR 1 CYCLE
17 03235 102400          SUB     0,0             ; CLEAR FOR REGISTER
18 03236 062022          DOB     0,.MTA          ; CLEAR MA REGISTER
19 03237 063022          DOC     0,.MTA          ; CLEAR WORD COUNT
20 03240 020235          LDA     0,C5X
21 03241 061122          DOAS    0,.MTA          ;WRITE
22 03242 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
23 03243 063522          SKPBZ   .MTA
24 03244 000776          JMP     .-2
25 03245 064422          DIA     1,.MTA          ;READ STATUS
26 03246 030147          LDA     2,C1000
27 03247 147414          AND#    2,1,SZR
28 03250 000407          JMP     ENDT            ;END TAPE
29 03251 125000 ETT.1:   MOV     1,1             ; TEST THE STATUS WORD
30 03252 125000          MOV     1,1
31 03253 125000          MOV     1,1             ; IF ERROR PROCESS IT
32 03254 125000          MOV     1,1
33 03255 006277          JSR     @LOOP.          ; TEST FOR ERROR LOOP
34 03256 000756          JMP     ETTA+1
35
36 03257 006207 ENDT:    MESSAGE
37 03260 004414          MENDT
38 03261 002401          JMP     @ETTB
39 03262 000000 ETTB:    0
40
41 03263 000000 RINWL:   0
42 03264 000000 INWLD:   0
43 03265 060022 CDCM:    060022
44                       ;WRITE LOCK TEST
45
46 03266 006264 WLT:     JSR     @INTIL
47 03267 004405          JSR     INWLT
48 03270 006207          MESSAGE
49 03271 003514          MSGTC
50 03272 006226          JSR     @IODT?
51 03273 000777          JMP     .-1
52 03274 054767 INWLT:   STA     3,RINWL
53 03275 060222          NIOC    .MTA            ; CLEAR CONTROLLER
54 03276 006275          JSR     @SETP1
55 03277 006317          REWIND
56 03300 006265 STA1:    STATUS
57 03301 100205          100205
58 03302 000403          JMP     STA2            ;RING IN OR OTHER TROUBLE
59 03303 101000          MOV     0,0
60 03304 000405          JMP     STA3            ;RING OUT
```

```
 0042 MT110
01 03305 006265 STA2:   STATUS
02 03306 100201         100201
03 03307 000771         JMP     STA1            ;RING OUT OR OTHER TROUBLE
04 03310 101000         MOV     0,0
05 03311 006265 STA3:   STATUS                  ;RING IN
06 03312 100201         100201                  ;SKP+2 IF RING IN
07 03313 000422         JMP     INWLA
08 03314 101000         MOV     0,0
09 03315 006207         MESSAGE                 ;RING IN
10 03316 004426         INWLB
11 03317 020157         LDA     0,D3000
12 03320 040744         STA     0,INWLD
13 03321 006265 STA4:   STATUS
14 03322 100201         100201
15 03323 000412         JMP     INWLA           ;GOT RING OUT OR UNRDY
16 03324 101000         MOV     0,0
17 03325 006224         WAIT
18 03326 000136         C100
19 03327 006313         JSR     @ITEST          ; TEST FOR CONSOLE INPUT
20 03330 014734         DSZ     INWLD
21 03331 000770         JMP     STA4
22 03332 006207         MESSAGE                 ;WAITED LONG TIME
23 03333 004471         INWLH
24 03334 002727         JMP     @RINWL
25 03335 020162 INWLA:  LDA     0,M4
26 03336 040726         STA     0,INWLD
27 03337 060222 INWLF:  NIOC    .MTA            ; CLEAR CONTROLLER
28 03340 006317         REWIND
29 03341 006265         STATUS
30 03342 100205         100205
31 03343 000774         JMP     INWLF           ;RING IN OR OTHER
32 03344 101000         MOV     0,0
33 03345 006267         WRITE                   ;OK RING OUT
34 03346 000102         102
35 03347 063522         SKPBZ   .MTA
36 03350 000777         JMP     .-1
37 03351 006265         STATUS
38 03352 110205         110205
39 03353 006300         EHALT                   ;NOT ILLEGAL OR OTHER ERROR
40 03354 000277         LOOP.                   ;AC1-EXPECTED STATUS,AC0-ACTUAL STATUS
41 03355 010707         ISZ     INWLD           ;ILLEGAL
42 03356 000761         JMP     INWLF
43 03357 002704         JMP     @RINWL
44       000001         .NOMAC  X
45 03360 106615 MSG1:   .TXTE   !<15><15><12>MOUNT A WRITE ENABLED SCRATCH TAPE, HIT ANY
46 03407 045640 KEY TO CONTINUE!
47 03420 005215 MSG2:   .TXTE   !<15><12>PASS !
48 03424 005215 MSGUN:  .TXTE   !<15><12>DRIVE UNIT #: !
49 03435 005215 MSGDC:  .TXTE   !<15><12>ENTER DEVICE CODE 22 OR 62.  !
50 03455 005215 MSG5:   .TXTE   !<15><12>SET SWITCH REGISTER TO DESIRED VALUE,
51 03501 054724 TYPE CR TO CONTINUE.  !
52 03514 005215 MSGTC:  .TXTE   !<15><12>TEST COMPLETE, WAITING IN ODT.  !
53 03535 005215 MSGWT:  .TXTE   !<15><12>WAITING IN ODT ON ERROR, TYPE A 'P'
54 03560 147724 TO CONTINUE.  !
55 03567 005215 MSGS1:  .TXTE   !<15><12>EXPECTED STATUS = !
56 03602 120254 MSGS2:  .TXTE   !, ACTUAL STATUS = !
57 03614 120254 MSGS3:  .TXTE   !, PC = !
58 03620 005215 MSGLP:  .TXTE   !<15><12>LOOPING ON ERROR!
59 03632 005215 MSGPC:  .TXTE   !<15><12>PERCENT ERROR RATE = !
60 03646 005215 EMS01:  .TXTE   !<15><12>SELD LINE NOT RESET BY IORST!
```

```
 0043 MT110
01 03666 005215 EMG02:  .TXTE   !<15><12>BUSY FLIP-FLOP NOT RESET ERROR!
02 03707 005215 EMG03:  .TXTE   !<15><12>BUSY FLIP-FLOP NOT RESET BY IORST!
03 03731 005215 EMG04:  .TXTE   !<15><12>BUSY FLIP-FLOP NOT SET ERROR!
04 03751 005215 EMG05:  .TXTE   !<15><12>DONE FLIP-FLOP NOT RESET ERROR!
05 03772 005215 EMG06:  .TXTE   !<15><12>DONE FLIP-FLOP NOT SET ERROR!
06 04012 005215 EMG07:  .TXTE   !<15><12>SEND CLOCK BIT ON TOO LONG ERROR!
07 04034 005215 EMG10:  .TXTE   !<15><12>FIRST CHARACTER TIME OUT ERROR!
08 04055 005215 EMG11:  .TXTE   !<15><12>DATA TRANSFER TIME OUT ERROR!
09 04075 005215 EMG12:  .TXTE   !<15><12>NO INTERRUPT ERROR!
10 04110 005215 EMG13:  .TXTE   !<15><12>ILLEGAL INTERRUPT WITH MASK BIT SET,
11 04133 046640  MASK = !
12 04140 005215 EMG14:  .TXTE   !<15><12>MTU SELECT ERROR, DIB COMMAND = !
13 04162 005215 EMG15:  .TXTE   !<15><12>MA REGISTER NOT RESET BY IORST!
14 04203 005215 EMG16:  .TXTE   !<15><12>MA REGISTER SETTING ERROR!
15 04221 005215 EMG17:  .TXTE   !<15><12>DATA TRANSFER ERROR, !
16 04235 005215 EMG20:  .TXTE   !<15><12>INTA DEVICE CODE ERROR<15><12>INTA
17 04255 142504 DEVICE CODE = !
18 04265 005215 EMG21:  .TXTE   !<15><12>MA REGISTER COUNTING ERROR<15><12>
19 04304 147507 GOOD VALUE = !
20 04313 005215 MSGCE:  .TXTE   !<15><12>DATA COMPARE ERROR<15><12>!
21 04327 147507 MSGWD:  .TXTE   !GOOD WORD = !
22 04336 120254 MSGBW:  .TXTE   !, BAD WORD = !
23 04345 120254 MSGWA:  .TXTE   !, MEMORY ADDRESS = !
24 04357 120254 MGDOB:  .TXTE   !, DOB = !
25 04364 120254 MGDIB:  .TXTE   !, DIB = !
26 04371 120254 MSGBV:  .TXTE   !, BAD VALUE = !
27 04401 120254 MGUDC:  .TXTE   !, UNIT DEVICE CODE = !
28 04414 005215 MENDT:  .TXTE   !<15><12>END TAPE!
29 04422 005215 MCYC:   .TXTE   !<15><12>CYCLE!
30 04426 005215 INWLB:  .TXTE   !<15><12>MOVE CARTRIDGE SWITCH TO THE SAFE POSITION. DONT STOP THE PROGRAM. !
31 04471 005215 INWLH:  .TXTE   !<15><12>WRITE LOCK BIT OFF: 100201 !
32 04510 005215 PRB:    .TXTE   !<15><12>MOVE SWITCH OFF THE SAFE POSITION !
33                      ;LONGITUDINAL PARITY DATA LIST
34
35 04533 004000 LST79:  4000                    ;RB2-P
36 04534 010000         10000                   ;3-P
37 04535 020000         20000                   ;4-P
38 04536 000400         400                     ;5-P
39 04537 001000         1000                    ;6-P
40 04540 002000         2000                    ;7-P
41 04541 030000         30000                   ;2-3
42 04542 014000         14000                   ;3-4
43 04543 006000         6000                    ;4-5
44 04544 003000         3000                    ;5-6
45 04545 001400         1400                    ;6-7
46 04546 000600         600                     ;7-2
47 04547 000000         0
48
49
50 04550 100000         100000                  ;0-P
51 04551 040000         40000                   ;1-P
52 04552 140000         140000                  ;0-1
53 04553 120000         120000                  ;0-2
54 04554 060000         60000                   ;1-2
55 04555 000000         0
56 04556 000000         0
57
58                      ;BAD TAPE STATUS WRITE BUFFER
59
60 04557 020001 BTBF:   020001                  ;1
```

```
 0044 MT110
01 04560 010002          010002                  ;2
02 04561 004004          004004                  ;3
03 04562 002010          002010                  ;4
04 04563 000000          000000                  ;5
05 04564 000002          000002                  ;6
06 04565 000420          000420                  ;7
07 04566 020040          020040                  ;8
08
09                       ;CRC CHECK WRITE BUFFERS
10
11 04567 000200 PAT1:    000200
12 04570 000200          000200
13 04571 153600          153600
14 04572 177474 PAT2:    177474
15 04573 136074          136074
16 04574 112074          112074
17 04575 052652 PAT3:    052652
18 04576 076232          076232
19 04577 037232          037232
20 04600 167141 PAT4:    167141
21 04601 000200          000200
22 04602 063600          063600
23
24 04603 000077 PAT5:    77
25 04604 000077          77
26 04605 000077          77
27 04606 037400 PAT6:    037400
28 04607 037400          037400
29 04610 037400          037400
30 04611 012452 PAT7:    012452
31 04612 012452          012452
32 04613 012452          012452
33 04614 012452 PAT8:    012452
34 04615 012477          012477
35 04616 012477          012477
36 04617 054420          STA     3,LOOPR
37 04620 176520          SUBZL   3,3
38 04621 000403          JMP     .+3
39 04622 054415 ENTER:   STA     3,LOOPR         ;LOOP ITERATE RETURN
40 04623 034407          LDA     3,ITR           ;THIS ROUTINE INITIALIZES
41 04624 054407          STA     3,ITRCT         ;EACH TEST
42 04625 176400          SUB     3,3
43 04626 054406          STA     3,ESWIT
44 04627 054406          STA     3,ERRCT
45 04630 060222          NIOC    .MTA            ; CONTROLLER RESET
46 04631 002406          JMP     @LOOPR
47
48 04632 000012 ITR:     12
49 04633 000000 ITRCT:   0
50 04634 000000 ESWIT:   0
51 04635 000000 ERRCT:   0
52 04636 000000 RETURN:  0
53 04637 000000 LOOPR:   0
54
55 04640 054776 CYCLE:   STA     3,RETURN        ;END OF TEST ITERATION
56 04641 050531          STA     2,SAV2          ;ROUTINE
57 04642 044527          STA     1,SAV1          ;SAVE THE ACS'
58 04643 040525          STA     0,SAV0
59 04644 014767          DSZ     ITRCT
60 04645 000433          JMP     CYCTS           ;NOT 10 TIMES ITERATED
```

```
 0045 MT110
01
02 04646 034764          LDA     3,ITR           ;RESET ITERATION CNTR
03 04647 054764          STA     3,ITRCT
04 04650 036205          LDA     3,@ISWR?EG      ; GET SWITCH VALUES
05 04651 030763          LDA     2,ESWIT         ;IF SWITCH 3=(1)
06 04652 175120          MOVZL   3,3             ;AND A ERROR HAS OCCURED
07 04653 175100          MOVL    3,3             ;THE ERROR RATE WILL
08 04654 151005          MOV     2,2,SNR         ;BE PRINTED
09 04655 000414          JMP     NOEX
10 04656 177103          ADDL    3,3,SNC
11 04657 000417          JMP     PCENT
12
13 04660 006207          MESSAGE                 ; PRINT PERCENT MESSAGE
14 04661 003632          MSGPC
15 04662 024753          LDA     1,ERRCT
16 04663 030143          LDA     2,C144
17 04664 006227          JSR     @IMLT?          ;AC1XAC2
18 04665 030745          LDA     2,ITR
19 04666 006230          JSR     @IDVI?          ;AC0-1/AC2
20 04667 006212          TYPZD                   ;PRINT VALUE
21 04670 000406          JMP PCENT
22
23 04671 020477 NOEX:    LDA     0,SAV0          ;NORMAL EXIT,NO ERR
24 04672 024477          LDA     1,SAV1
25 04673 030477          LDA     2,SAV2
26 04674 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
27 04675 002741          JMP     @RETURN
28
29
30 04676 102400 PCENT:   SUB     0,0
31 04677 040736          STA     0,ERRCT         ;RESET ERROR COUNT
32 04700 020470 CYCTS:   LDA     0,SAV0          ;RESTORE ACS
33 04701 024470          LDA     1,SAV1
34 04702 030470          LDA     2,SAV2
35 04703 060222          NIOC    .MTA            ; CLEAR CONTROLLER
36 04704 006313          JSR     @ITEST          ; TEST FOR CONSOLE INPUT
37 04705 034727          LDA     3,ESWIT
38 04706 175005          MOV     3,3,SNR         ; TEST FOR ERROR SWITCH
39 04707 002730          JMP     @LOOPR          ; IF NOT SET, LOOP
40 04710 036205          LDA     3,@ISWR?EG      ; GET SWITCH REGISTER
41 04711 177113          ADDL#   3,3,SNC         ;SWITCH 1
42 04712 002725          JMP     @LOOPR          ;(0)=LOOP ROUTINE
43 04713 002723          JMP     @RETURN         ;(1)=PROCEED TO NEXT TEST
44
45 04714 054722 SPER1:   STA     3,RETURN        ; SAVE RETURN ADDRESS
46 04715 176520          SUBZL   3,3             ; SET BIT 15
47 04716 000406          JMP     EERR            ; GO SET CONTROL FLAG
48
49 04717 054717 SPER2:   STA     3,RETURN        ; SAVE RETURN ADDRESS
50 04720 176620          SUBZR   3,3             ; SET BIT 0
51 04721 000403          JMP     EERR            ; GO SET CONTROL FLAG
52
53 04722 054714 ERR:     STA     3,RETURN        ;ERROR SUBROUTINE
54 04723 176400          SUB     3,3             ; CLEAR REGISTER
55 04724 054473 EERR:    STA     3,EFLAG         ; SET ERROR CONTROL FLAG
56 04725 050445          STA     2,SAV2
57 04726 044443          STA     1,SAV1
58 04727 040441          STA     0,SAV0
59
60 04730 034704          LDA     3,ESWIT
```

```
 0046 MT110
01 04731 175005          MOV     3,3,SNR
02 04732 000413          JMP     ERR1
03 04733 032205          LDA     2,@ISWR?EG      ; GET SWITCH REGISTER
04 04734 024145          LDA     1,C200          ; GET MASK BIT
05 04735 133414          AND#    1,2,SZR         ; TEST FOR PRINT ALL
06 04736 004435          JSR     EPRNT           ; IF SET, PRINT STATUS
07
08 04737 030433 ERET:    LDA     2,SAV2          ;RESTORE ACS
09 04740 024431          LDA     1,SAV1
10 04741 020427          LDA     0,SAV0
11 04742 010673          ISZ     ERRCT           ;COUNT
12 04743 101000          MOV     0,0             ;ERRORS,I/O RESET
13 04744 002672          JMP     @RETURN         ;EXIT
14
15 04745 004426 ERR1:    JSR     EPRNT           ; GO PRINT STATUS
16 04746 032205          LDA     2,@ISWR?EG      ; GET SWITCH REGISTER
17 04747 020147          LDA     0,C1000         ; GET MASK BIT
18 04750 113415          AND#    0,2,SNR         ; TEST FOR WAIT ON ERROR
19 04751 000404          JMP     EPRINT          ;PRINT ERROR DATA
20 04752 006207          MESSAGE                 ; PRINT WAIT MESSAGE
21 04753 003535          MSGWT
22 04754 006226          JSR     @IODT?          ; EXIT TO ODT
23
24 04755 032205 EPRINT:  LDA     2,@ISWR?EG      ; GET SWREG VALUE
25 04756 024156          LDA     1,C40000        ; GET MASK BIT
26 04757 133404          AND     1,2,SZR         ; TEST FOR LOOP ON ERROR
27 04760 000757          JMP     ERET            ; IF NO LOOP, RETURN
28 04761 020655          LDA     0,RETURN        ; GET RETURN ADDRESS
29 04762 040652          STA     0,ESWIT         ; SET THE ERROR SWITCH
30 04763 006207          MESSAGE                 ; PRINT LOOPING MESSAGE
31 04764 003620          MSGLP
32 04765 024645          LDA     1,ITR           ; GET ITERATION COUNT
33 04766 044645          STA     1,ITRCT         ; SET COUNTER
34 04767 000750          JMP     ERET            ; PROCESS RETURN
35
36 04770 000000 SAV0:    0
37 04771 000000 SAV1:    0
38 04772 000000 SAV2:    0
39
40 04773 054423 EPRNT:   STA     3,MSAV          ; SAVE THE RETURN
41 04774 034423          LDA     3,EFLAG         ; GET CONTROL FLAG
42 04775 175212          MOVR#   3,3,SZC         ; TEST FOR BIT 15 SET
43 04776 000441          JMP     SPERT           ; IF SET, PROCESS ERROR
44 04777 175112          MOVL#   3,3,SZC         ; TEST FOR BIT 0 SET
45 05000 000420          JMP     CPERT           ; IF SET, PROCESS ERROR
46 05001 006207          MESSAGE                 ; PRINT EXPECTED STATUS
47 05002 003567          MSGS1
48 05003 024766          LDA     1,SAV1          ; GET EXPECTED STATUS
49 05004 006214          TYPEO                   ; PRINT THE STATUS WORD
50 05005 006207          MESSAGE                 ; PRINT ACTUAL STATUS
51 05006 003602          MSGS2
52 05007 024761          LDA     1,SAV0          ; GET THE ACTUAL STATUS
53 05010 006214          TYPEO                   ; PRINT THE STATUS WORD
54 05011 006207 EPRT1:   MESSAGE                 ; PRINT PC
55 05012 003614          MSGS3
56 05013 024623          LDA     1,RETURN        ; GET RETURN ADDRESS
57 05014 006214          TYPEO                   ; PRINT THE PC WORD
58 05015 002401          JMP     @MSAV           ; RETURN TO CALL
59
60 05016 000000 MSAV:    0
```

```
 0047 MT110
01 05017 000000 EFLAG:  0
02
03              ; PRINT COMPARE ERROR
04
05 05020 006207 CPERT:  MESSAGE                 ; PRINT ERROR MESSAGE
06 05021 004313         MSGCE
07 05022 006207         MESSAGE                 ; PRINT GOOD WORD =
08 05023 004327         MSGWD
09 05024 024744         LDA     1,SAV0          ; GET GOOD WORD
10 05025 006214         TYPEO                   ; PRINT IT IN OCTAL
11 05026 006207         MESSAGE                 ; PRINT BAD WORD =
12 05027 004336         MSGBW
13 05030 024741         LDA     1,SAV1          ; GET BAD WORD
14 05031 006214         TYPEO                   ; PRINT IT IN OCTAL
15 05032 006207         MESSAGE                 ; PRINT MEMORY ADDRESS =
16 05033 004345         MSGWA
17 05034 024736         LDA     1,SAV2          ; GET MEMORY ADDRESS
18 05035 006214         TYPEO                   ; PRINT IT IN OCTAL
19 05036 000753         JMP     EPRT1           ; GO PRINT PC =
20
21              ; PRINT ALL OTHER ERROR MESSAGES
22
23 05037 030520 SPERT:  LDA     2,MSGTB         ; GET MESSAGE TABLE ADDRESS
24 05040 024732         LDA     1,SAV2          ; GET MESSAGE NUMBER
25 05041 133000         ADD     1,2             ; SET TO TABLE ADDRESS
26 05042 035000         LDA     3,0,2           ; GET THE MESSWAGE ADDRESS
27 05043 020117         LDA     0,C13           ; GET FOR PRINT CONTROL
28 05044 106432         SUBZ#   0,1,SZC         ; TEST FOR < 13
29 05045 000405         JMP     SPRT1           ; IF NOT < 13, FIND PRINT OUT
30 05046 054402         STA     3,PMSG1         ; SAVE THE MESSAGE ADDRESS
31 05047 006207         MESSAGE                 ; PRINT REQUESTED MESSAGE
32 05050 000000 PMSG1:  0                       ; MESSAGE ADDRESS
33 05051 000740         JMP     EPRT1           ; GO PRINT PC =
34              ; TEST FOR PRINT INTERRUPT MASK ERROR
35
36 05052 106414 SPRT1:  SUB#    0,1,SZR         ; TEST FOR MESSAGE 13
37 05053 000407         JMP     SPRT2           ; IF NOT 13, TEST NEXT
38 05054 054402         STA     3,PMSG2         ; SAVE THE MESSAGE ADDRESS
39 05055 006207         MESSAGE                 ; PRINT MASK ERROR
40 05056 000000 PMSG2:  0                       ; MESSAGE ADDRESS
41 05057 024712         LDA     1,SAV1          ; GET THE MASK WORD
42 05060 006214         TYPEO                   ; PRINT IT IN OCTAL
43 05061 000730         JMP     EPRT1           ; GO PRINT PC =
44
45              ; TEST FOR PRINT MTU SELECT ERROR
46
47 05062 101400 SPRT2:  INC     0,0             ; SET = 14
48 05063 106414         SUB#    0,1,SZR         ; TEST FOR MESSAGE 14
49 05064 000407         JMP     SPRT3           ; IF NOT 14, TEST NEXT
50 05065 054402         STA     3,PMSG3         ; SAVE THE MESSAGE ADDRESS
51 05066 006207         MESSAGE                 ; PRINT MTU SELECT ERROR
52 05067 000000 PMSG3:  0                       ; MESSAGE ADDRESS
53 05070 024701         LDA     1,SAV1          ; GET DIB COMMAND
54 05071 006214         TYPEO                   ; PRINT IT IN OCTAL
55 05072 000717         JMP     EPRT1           ; GO PRINT PC =
56
57              ; TEST FOR MA REGISTER ERROR
58
59 05073 020123 SPRT3:  LDA     0,C17           ; SET FOR MESSAGE 17
60 05074 106432         SUBZ#   0,1,SZC         ; TEST FOR < 17
```

```
 0048 MT110
01 05075 000415         JMP     SPRT4           ; IF NOT < 17, TEST NEXT
02 05076 054402         STA     3,PMSG4         ; SAVE THE MESSAGE ADDRESS
03 05077 006207         MESSAGE                 ; PRINT THE REQUESTED MESSAGE
04 05100 000000 PMSG4:  0                       ; MESSAGE ADDRESS
05 05101 006207         MESSAGE                 ; PRINT DOB = MESSAGE
06 05102 004357         MGDOB
07 05103 024665         LDA     1,SAV0          ; GET DATA SENT
08 05104 006214         TYPEO                   ; PRINT IT IN OCTAL
09 05105 006207         MESSAGE                 ; PRINT DIB = MESSAGE
10 05106 004364         MGDIB
11 05107 024662         LDA     1,SAV1          ; GET DATA INPUT
12 05110 006214         TYPEO                   ; PRINT IT IN OCTAL
13 05111 000700         JMP     EPRT1           ; GO PRINT PC =
14
15              ; TEST FOR PRINT DATA TRANSFER ERROR
16
17 05112 106414 SPRT4:  SUB#    0,1,SZR         ; TEST FOR MESSAGE 17
18 05113 000415         JMP     SPRT5           ; IF NOT SET, TEST NEXT
19 05114 054402         STA     3,PMSG5         ; SAVE MESSAGE ADDRESS
20 05115 006207         MESSAGE                 ; PRINT DATA TRANSFER ERROR
21 05116 000000 PMSG5:  0                       ; MESSAGE ADDRESS
22 05117 006207         MESSAGE                 ; PRINT GOOD WORD =
23 05120 004327         MSGWD
24 05121 024650         LDA     1,SAV1          ; GET GOOD DATA WORD
25 05122 006214         TYPEO                   ; PRINT IT IN OCTAL
26 05123 006207         MESSAGE                 ; PRINT BAD WORD =
27 05124 004336         MSGBW
28 05125 024643         LDA     1,SAV0          ; GET BAD DATA WORD
29 05126 006214         TYPEO                   ; PRINT IT IN OCATAL
30 05127 000662         JMP     EPRT1           ; GO PRINT PC =
31
32              ; TEST FOR PRINT INTA DEVICE CODE ERROR
33
34 05130 101400 SPRT5:  INC 0,0                 ; SET FOR MESSAGE 20
35 05131 106414         SUB#    0,1,SZR         ; TEST FOR MESSAGE 20
36 05132 000413         JMP     SPRT6           ; IF NOT 20, DO 21
37 05133 054402         STA     3,PMSG6         ; SAVE MESSAGE ADDRESS
38 05134 006207         MESSAGE                 ; PRINT INTA DEVICE CODE ERROR
39 05135 000000 PMSG6:  0                       ; MESSAGE ADDRESS
40 05136 024632         LDA     1,SAV0          ; GET INTA DEVICE CODE
41 05137 006215         TYPZO                   ; PRINT IT IN OCTAL
42 05140 006207         MESSAGE                 ; PRINT UNIT DEVICE CODE =
43 05141 004401         MGUDC
44 05142 024627         LDA     1,SAV1          ; GET DEVICE CODE
45 05143 006215         TYPZO                   ; PRINT IT IN OCTAL
46 05144 000645         JMP     EPRT1           ; GO PRINT PC =
47
48              ; PRINT MA REGISTER COUNTING ERROR
49
50 05145 054402 SPRT6:  STA     3,PMSG7         ; SAVE THE MESSAGE ADDRESS
51 05146 006207         MESSAGE                 ; PRINT MA COUNTING ERROR
52 05147 000000 PMSG7:  0                       ; MESSAGE ADDRESS
53 05150 024621         LDA     1,SAV1          ; GET GOOD VALUE
54 05151 006214         TYPEO                   ; PRINT IT IN OCTAL
55 05152 006207         MESSAGE                 ; PRINT BAD VALUE =
56 05153 004371         MSGBV
57 05154 024614         LDA     1,SAV0          ; GET BAD VALUE WORD
58 05155 006214         TYPEO                   ; PRINT IT IN OCTAL
59 05156 000633         JMP     EPRT1           ; GO PRINT PC =
60
```

```
 0049 MT110
01              ; MESSAGE TABLE
02
03 05157 005157 MSGTB:  .
04 05160 003646         EMG01
05 05161 003666         EMG02
06 05162 003707         EMG03
07 05163 003731         EMG04
08 05164 003751         EMG05
09 05165 003772         EMG06
10 05166 004012         EMG07
11 05167 004034         EMG10
12 05170 004055         EMG11
13 05171 004075         EMG12
14 05172 004110         EMG13
15 05173 004140         EMG14
16 05174 004162         EMG15
17 05175 004203         EMG16
18 05176 004221         EMG17
19 05177 004235         EMG20
20 05200 004265         EMG21
21
22 05201 000000         0                       ; RETURN ADDRESS
23 05202 054777 TSTIN:  STA     3,TSTIN-1       ; SAVE THE RETURN ADDRESS
24 05203 040422         STA     0,TSAV0         ; SAVE REGISTER 0
25 05204 034221         LDA     3,IOM?OD        ; GET I/O MOD
26 05205 175005         MOV     3,3,SNR         ; TEST FOR FLAG SET
27 05206 000405         JMP     TSTI1           ; IF NOT SET, GET CONSOLE
28 05207 006221         JSR     @IOM?OD         ; GET I/O INPUT
29 05210 000000         0                       ; INPUT REQUEST
30 05211 000404         JMP     TSTI2           ; IF INPUT READY, TEST IT
31 05212 000411         JMP     TSTI3           ; NOT READY, EXIT
32 05213 063610 TSTI1:  SKPDN   TTI             ; TEST FOR TTY READY
33 05214 000407         JMP     TSTI3           ; NOT READY, EXIT
34 05215 006206 TSTI2:  JSR     @IINP?          ; GO TEST FOR SW REG
35 05216 020123         LDA     0,CONO          ; GET CONTROL O CHARACTER
36 05217 116400         SUB     0,3             ; TEST FOR ODT ENTRY
37 05220 020405         LDA     0,TSAV0         ; RESTORE REGISTER 0
38 05221 175005         MOV     3,3,SNR         ; TEST FOR ODT ENTRY
39 05222 006226         JSR     @IODT?          ; IF YES, ENTER ODT
40 05223 020402 TSTI3:  LDA     0,TSAV0         ; RESTORE REGISTER 0
41 05224 002755         JMP     @TSTIN-1        ; RETURN TO CALL
42
43 05225 000000 TSAV0:  0
44
45                      T?TY0
46                      S?WPK
47
48                      O?DTP
49                      O?DTD   2
50                      T?MER
```

```
 0050 MT110
01                      N?MOV
02 06757 000000 EGGS:   0
03 06760 000000         0
04 06761 000000         0
05 06762 000000         0
06 06763 000000         0
07 06764 000000 SWREG:  0
08 06765 005215 DIRT:   .TXTE   !<15><12)..C.S.I.  MT110D DIAG.(KENNEDY 1/4") REV. 00 !
09       027056
10       141456
11       051456
12       144456
13       120056
14       046640
15       130724
16       030261
17       120104
18       144504
19       043501
20       024056
21       142513
22       047116
23       042305
24       120131
25       127661
26       021264
27       120251
28       142722
29       027126
30       030240
31       120060
32       000000
33 07016 000000 LAST:   0
34       000525 RES?T=A0
35                      .END
**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

A0	000525	12/31	50/34				
A1	000531	12/38	32/27				
A10	000761	15/51					
A11	000773	16/04					
A11A	001004	16/10	16/13				
A11B	001007	16/12	16/16				
A12	001010	16/20					
A13	001020	16/31					
A14	001030	16/42					
A14A	001045	17/02					
A14B	001055	17/13					
A14C	001065	17/18	17/21				
A14D	001070	17/20	17/24				
A14.1	001032	16/44	16/58				
A15	001071	17/28					
A15E	001105	17/34	17/41				
A16	001106	17/45					
A2	000541	12/49					
A20	001117	17/58					
A20E	001146	18/07	18/17	18/23			
A20.1	001121	17/60					
A21	001147	18/27					
A21A	001157	18/38					
A21B	001170	18/50					
A22	001204	19/05					
A22E	001237	19/13	19/29	19/35			
A22.1	001210	19/09	19/44				
A22.2	001212	19/06	19/12	19/43			
A23	001250	19/40	19/48				
A23E	001303	20/02	20/11	20/17			
A23.1	001254	19/52	20/26				
A23.2	001256	19/49	19/55	20/25			
A24	001314	20/22	20/30				
A25	001333	20/48					
A26	001352	21/09					
A26A	001344	20/60					
A28	001361	21/18					
A28E	001400	21/28	21/33				
A3	000546	12/57					
A30	001401	21/37					
A30E	001424	21/42	21/51	21/57			
A31	001425	21/60					
A32	001437	22/15					
A32E	001463	22/23	22/29	22/36			
A33	001464	22/41					
A33E	001537	22/54	22/59	23/07	23/11	23/21	23/26
A33.1	001470	22/45	23/35				
A33.2	001473	22/42	22/48	23/34			
A34	001550	23/31	23/40				
A34E	001623	23/53	23/58	24/06	24/10	24/19	24/24
A34.1	001554	23/44	24/33				
A34.2	001557	23/41	23/47	24/32			
A35	001634	24/29	24/37				
A35E	001676	24/46	24/54	24/58	25/08	25/13	
A36	001677	25/16					
A36E	001741	25/25	25/33	25/37	25/48	25/52	
A39	001742	25/53					
A39E	001772	26/01	26/10	26/17			
A4	000553	13/05					

C1004	000152	8/28	26/05					
C100.	000103	8/16	36/13	36/30				
C101	000137	8/17	22/30					
C102	000140	8/18	21/53					
C103	000141	8/19	27/06					
C104	000142	8/20	26/39					
C1042	000071	7/37	14/25	15/10				
C10?0	005665	49/46	49/47					
C11	000115	7/58						
C1100	000100	7/44	20/40					
C12	000116	7/59	18/30					
C1252	000101	7/45	13/07					
C13	000117	7/60	18/55	47/27				
C14	000120	8/01	13/35					
C144	000143	8/21	45/16					
C15	000121	8/02	11/04	14/15				
C150	000144	8/22	30/56					
C16	000122	8/03	13/06	13/19	13/58	14/29	14/40	14/51
		15/02	15/19					
C1600	000075	7/41	33/47					
C17	000123	8/04	11/05	47/59				
C1777	000077	7/43	33/53					
C1X	000233	9/30	16/23	33/22	36/03	36/28		
C2	000106	7/51	12/50	16/13	16/22	17/16	18/10	32/02
C20	000124	8/05	18/41	33/28	40/07			
C200	000145	8/23	28/34	29/11	29/48	30/24	46/04	
C2000	000150	8/26	8/33					
C201	000146	8/24	36/36					
C20K	000151	8/27	17/48	36/16				
C20.	000125	8/06	35/47	36/55	38/45	39/43	40/05	
C21	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				
C2104	000072	7/38	14/36	15/11				
C22	000127	8/08	33/35					
C2MS	000150	8/33						
C3	000107	7/52	15/57					
C30	000130	8/09	35/31					
C3747	000155	8/31						
C4	000110	7/53	15/44					
C40	000131	8/10	18/53	33/24	35/25			
C4000	000156	8/32	46/25					
C4210	000073	7/39	14/47	15/12				
C421T	000074	7/40	14/58	15/13				
C4K	000153	8/29	32/18	40/55				
C5	000111	7/54	12/58	15/38	16/08			
C5252	000102	7/46	13/20	13/32				
C5X	000235	9/32	33/26	39/28	41/20			
C6	000112	7/55	17/21	18/03	19/15	19/58	21/02	21/24
		23/03	24/02	24/50	25/29			
C60	000132	8/11	36/52					
C62	000133	8/12	33/38					
C7	000113	7/56	40/24					
C70	000134	8/13	16/42	40/02				
C77	000135	8/14	12/13	13/43	38/20	39/20		
C7700	000154	8/30	38/30	39/31				
C7X	000237	9/34	33/30	38/02				
CAC?0	005252	49/46						

```
INM?  006064      49/47
INP?1 005742      49/47
INP?I 005745      49/47
INP?J 005750       9/06   49/47
INP?K 006400      49/50
INP?Q 006142      49/47   49/50
INP?R 006150      49/47   49/50
INR?  006012      49/47   49/50
INR?1 006411      49/50
INR?K 006410      49/50
INR?O 006412      49/50
INST  000242       9/37   38/29   38/43   39/30   39/38
INS?  006154      49/47
INS?0 006144      49/47
INS?1 006145      49/47
INS?2 006146      49/47
INS?3 006147      49/47   49/50
INS?A 006143      49/47   49/50
INS?V 006124      49/47   49/50
INS?X 006205      49/50
INT10 002525      34/06   34/21
INT11 002533      34/08   34/12
INT12 002535      34/11   34/14
INT13 002545      34/17   34/22
INTIL 000264      10/03   12/34   41/08   41/46
INTR  000010       7/05    7/15
INT?  005630      49/46   49/47
INT?E 005737      49/47
INWLA 003335      42/07   42/15   42/25
INWLB 004426      42/10   43/30
INWLD 003264      41/42   42/12   42/20   42/26   42/41
INWLF 003337      42/27   42/31   42/42
INWLH 004471      42/23   43/31
INWLT 003274      10/22   41/47   41/52
IN?PR 005741      49/47
IODT? 000226       7/17    8/58    9/22   34/22   41/13   41/50   46/22
                  49/39   49/50
IOM?O 000221       9/17   12/20   34/06   34/09   49/25   49/28   49/46
                  49/47
IPC7? 000231       9/25   12/25
IPDC? 000213       9/11   49/47   49/50
IPDE? 000212       9/10   10/60   11/02
IPOC? 000214       9/12   10/58   49/50
IRD   000301      10/16   10/43
ISEL  000306      10/21   10/47
ISPC  000305      10/20   10/39
ISPE1 000314      10/27   10/41
ISPE2 000315      10/28   10/42
ISTAT 000265      10/04   10/31
ISWR? 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
ITEST 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
ITIM? 000225       9/21
ITI?A 000222       9/18   10/57
ITI?D 000217       9/15   50/01
ITI?O 000216       9/14   10/56
```

```
MSGS  003455       34/05   42/50
MSGS1 003567       42/55   46/47
MSGS2 003602       42/56   46/51
MSGS3 003614       42/57   46/55
MSGTB 005157       47/23   49/03
MSGTC 003514       41/12   41/49   42/52
MSGUN 003424       33/13   42/48
MSGWA 004345       43/23   47/16
MSGWD 004327       43/21   47/08   48/23
MSGWT 003535       42/53   46/21
MUL?L 006734       50/02
MUL?P 006722       50/02
MUL?R 006733       50/02
MUL?S 006724       50/02
MUL?T 006721        9/23   50/02
M?ESS 004660  MC   50/01
NEST  000045        7/21   12/08
NOEX  004671       45/09   45/23
N?DIV 030316  MC   50/02
N?MDV 030177  MC   50/01
N?MUL 030207  MC   50/02
O10?1 006162       49/50
OBUFF 000063        7/31   18/09   19/21   20/03   30/37   30/52   34/39
                   37/17   37/43   39/36
OCTIN 006216       10/56   33/14   33/33
OD1?2 006157       49/50
OD1?5 006160       49/50
OD6?0 006156       49/50
OD7?  006364       49/50
ODA?  006163       49/50
ODA?C 006161       49/50
ODA?L 006306       49/50
ODB?E 006347       49/50
ODB?P 006226       49/50
ODD?B 006223       49/50
ODD?R 006330       49/50
ODD?T 006370       49/50
ODE?1 006353       49/50
ODE?2 006355       49/50
ODE?4 006362       49/50
ODE?Q 006366       49/50
ODI?N 006373       49/50
ODI?T 006365       49/50
ODL?C 006274       49/50
ODL?T 006372       49/50
ODO?C 006314       49/50
ODO?F 006371       49/50
ODO?K 000204        9/01   49/50
ODP?C 006172       49/50
ODR?T 006173       49/50
ODT?1 006217       49/50
ODT?2 006242       49/50
ODT?3 006376       49/50
ODT?I 006207       49/50
ODT?J 006212        9/22   49/50
ODT?K 006413       49/50
ODT?P 006375       49/50
ODU?A 006367       49/50
```

```
ODW?T 006245       49/50
OKRET 003176       40/31   40/36   40/38
ON    003201       40/27   40/34
O?DTD 000527  MC    5/51   49/49
O?DTP 022740  MC   49/48
P17?? 005561       49/46   49/47
P3??? 005247       49/46
PAC?0 005376       49/46
PAC?1 005523       49/46
PAC?2 005377       49/46
PA5?5 000203        8/60   31/36   32/14   32/16   32/23   33/05
PAT1  004567       44/11
PAT2  004572       44/14
PAT3  004575       44/17
PAT4  004600       44/20
PAT5  004603       44/24
PAT6  004606       44/27
PAT7  004611       44/30
PAT8  004614       44/33
PA?C1 005667       49/46
PA?C3 005666       49/46
PC1?0 005402       49/46
PC1?1 005250       49/46
PC1?2 005661       49/46   49/47
PC1?5 005662       49/46   49/47
PC4?0 005532       49/46
PC6?0 005401       49/46
PC7?  005533        9/25   49/46
PCDCM 000076        7/42   33/49
PCENT 004676       45/11   45/21   45/30
PCR?Y 005375       49/46
PDC?1 005335       49/46
PDC?2 005333       49/46
PDC?5 005320        9/11   49/46
PDE?C 005330        9/10   49/46
PETTA 000310       10/23   32/05
PINWL 000307       10/22   31/43
PLP?T 005460       49/46
PMSG1 005050       47/30   47/32
PMSG2 005056       47/38   47/40
PMSG3 005067       47/50   47/52
PMSG4 005100       48/02   48/04
PMSG5 005116       48/19   48/21
PMSG6 005135       48/37   48/39
PMSG7 005147       48/50   48/52
POC?T 005312        9/12   49/46
POINT 000246        9/44   19/07   22/43
PONTR 000255        9/54   19/50   23/42
PRB   004510       31/50   43/32
PSP?  005245       49/46
RB6?0 005723       49/46
RBCT  002330       30/23   31/13   31/16
RCHK  002613       34/38   34/59   34/60   35/02   35/19
READ  006301       10/43   23/01   23/60   24/48   25/27   26/03
RECTR 000244        9/39   28/11   28/14   28/48   28/51   29/25   29/28
                   30/02   30/05
RES?T 000525       49/47   50/34
RETUR 004636       44/52   44/55   45/27   45/43   45/45   45/49   45/53
```

```
                  42/37
STA?T 006151      49/47
STN?D 006551      50/01
SVE?0 006715      50/01
SVE?1 006716      50/01
SWREG 006764       9/05   50/07
SWTCH 003143      39/06   39/16   39/40   39/57
S?WPD 000050  MC    3/04   49/47
S?WPK 021330  MC   49/46
TAC?0 005525      49/46
TAC?C 005531      49/46
TCN?T 006565      50/01
TIM?1 006717      50/01
TIM?2 006720      50/01
TIM?A 006677      50/01
TIM?H 006711      50/01
TIM?L 006704      50/01
TIM?R 006642       9/21   50/01
TIN?1 005663      49/46
TIN?2 005664      49/46
TIN?A 005670       9/18   49/46
TIN?C 005534      49/46
TIN?D 005574       9/15   49/46
TIN?M 005631      49/46
TIN?N 005642      49/46
TIN?O 005570       9/14   49/46
TIN?Q 005577      49/46
TIN?R 005537      49/46
TIN?S 005604      49/46
TIN?W 005610      49/46
TIN?X 005536      49/46
TIN?Z 005600      49/46
TMP?  005404      49/46
TOD?T 005562      49/46   49/50
TO?DT 006377      49/50
TPR?T 005500      49/46
TPS?P 005421       9/16   49/46
TP??  006374      49/50
TSAV0 005225      49/24   49/37   49/40   49/43
TSI?  005541      49/46
TSTI1 005213      49/27   49/32
TSTI2 005215      49/30   49/34
TSTI3 005223      49/31   49/33   49/40
TSTIN 005202      10/26   49/23   49/41
TTY?  005443      49/46
TYPEA 006211      11/01   33/11
TYPED 006212      11/02   32/24
TYPEO 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
TYPZD 006212      10/60   45/20
TYPZO 006215      10/55   48/41   48/45
TYP?E 005424       9/09   49/46
TYP?R 005530      49/46
T?MER 013434  MC  49/50
T?TY0 016366  MC  49/45
WAIT  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
```

WCTR 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
WDCNT 000241 9/36 38/23 38/39 39/23 39/34
WEOF 006271 10/35 21/01 21/10 25/57 26/25 26/52
WLT 003266 10/24 41/46
WRITE 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
WRTNS 006273 10/37
WSW 002655 35/32 35/45 35/59
X70 000232 9/29 16/43 16/45 16/54 16/55 40/32
XBSP 002615 10/09 35/24
XCHK 002547 10/18 34/38
XCHK. 002560 34/47 34/54
XCLR 002577 10/17 35/06
XERAR 003164 40/01 40/12 40/15 40/16
XERAS 003144 10/05 39/60
XGEN 002763 10/07 37/39
XLD 002743 10/19 37/15
XRD 003015 10/16 38/17
XRD.1 003043 38/39 38/55
XSEL 003002 10/21 37/58
XSPC 002620 10/20 35/30
XSPC0 002622 35/26 35/32
XSPC1 002643 35/49 35/55
XSPRE 002653 35/38 35/46 35/53 35/56 35/57
XSTAT 003207 10/04 40/50
XWEOF 002722 10/08 36/50
XWNS 003067 10/10 39/05
XWRT 003072 10/06 39/07 39/15
XWRT. 003115 39/34 39/53
ZOC?T 005306 9/13 49/46
ZPO?T 005315 49/46
ZSU?P 005524 49/46
.200M 000067 7/35 12/41 14/14 15/56 35/50 36/22 36/33
.25MS 000065 7/33 37/60 38/05
.400M 000070 7/36
.50MS 000066 7/34 36/58 38/48 39/46
.MTA 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

		41/18	41/19	41/21	41/23	41/25	41/53	42/27
		42/35	44/45	45/35				
.RW	002661	10/50	36/10					
.RW2	002706	36/18	36/32	36/41				
.RWNS	002656	10/48	36/03					
.RWRE	002721	36/10	36/26	36/39	36/42	36/44		
.XBSP	002614	10/52	35/23					

```
 0001 UMTR     MACRO REV 06.70         11:16:43 08/10/82
01              ;
02              ;
03              ;
04              ;*******************************************************************************
05              ;
06              ; DECRIPTION: UNIVERSAL MAGNETIC TAPE RELIABILITY
07              ;
08              ;
09              ; CUSTOM SYSTEMS INC, 1982
10              ;*******************************************************************************
                        .TITL   UMTR
12       000001         X=1
13       000001         .NOMAC  X
14              ;1.     PROGRAM NAME:   UMTR.SR
15              ;
16              ;2.     REVISION HISTORY:
17              ;
18              ;               REV.      DATE
19  00422 000000        .REV     00.0   ;06/07/82
20              ;
21              ;3.     MACHINE REQUREMENTS
22              ;
23              ;       3.1     NOVA OR ECLIPSE FAMILY CPU'S
24              ;       3.2     MINIMUM OF 16K MEMORY
25              ;       3.3     MAG TAPE COUPLER (CONTROLLER) BOARD
26              ;       3.4     TELETYPE OR CRT AND CONTROLLER
27              ;       3.5     TAPE DRIVE (S)
28              ;4.     TEST REQUIREMENTS
29              ;
30              ;       N/A
31              ;
32              ;5.     SUMMARY
33              ;
34              ;       THE TAPE RELIABILITY PROGRAM IS A MAINTENANCE
35              ;       PROGRAM INTENDED TO VERIFY THE MAGNETIC TAPE
36              ;       SUB-SYTEM OPERATION.
37              ;
38              ;6.     RESTRICTIONS
39              ;
40              ;       ONLY THOSE TAPE DRIVES TO BE TESTED ARE TO
41              ;       BE ONLINE.  ALL ONLINE DRIVES MUST BE WRITE ENABLED.
```

```
!0002 UMTR
01              ;7.     PROGRAM DESCRIPTION/THEORY OF OPERATION
02              ;
03              ;       7.1     RANDOM RELIABILITY (SA 500)
04              ;
05              ;               THE RANDOM RELIABILITY TEST WRITES RANDOM
06              ;               LENGTH FILES.  EACH FILE CONSISTS OF FROM
07              ;               1 TO 7 RANDOM LENGTH, RANDOM PATTERN REC-
08              ;               ORDS.  THE RANDOM FILES ARE WRITTEN AND
09              ;               READ THE FULL LENGTH OF THE MEDIA.  IF
10              ;               MORE THAN ONE(1) TAPE DRIVE IS AVAILABLE,
11              ;               A UNIQUE RANDOM FILE WILL BE WRITTEN ON EACH
12              ;               UNIT SEQUENTIALILY.  WHEN EACH UNIT'S EOT
13              ;               SENSOR IS DETECTED, ITS ACCUMULATED
14              ;               HISTORY IS PRINTED AND THE UNIT IS COM-
15              ;               MANDED TO REWIND.  ALL WRITE ENABLED,
16              ;               READY TAPE UNITS WILL BE TESTED.  A UNIT
17              ;               CAN BE MADE READY AND WILL BE TESTED AFTER
18              ;               THE TEST HAS BEEN INITIATED.  IF A UNIT
19              ;               BECOMES NOT READY DURING THE TEST, ITS
20              ;               HISTORY WILL BE PRINTED AND THE UNIT
21              ;               WILL BE REMOVED FROM THE AVAILABLE UNITS
22              ;               LIST.  THE TEST WILL CONTINUE UNTIL STOPPED
23              ;               BY THE OPERATOR.
24              ;
25              ;       7.2     INTERCHANGE TEST, WRITE/READ (SA 501)
26              ;
27              ;               THE INTERCHANGE TEST IS USED TO VERIFY THE
28              ;               INTERCHANGABLITY OF THE TAPE UNITS.  THIS
29              ;               TEST GENERATES 200, 2000 WORD RECORDS OF
30              ;               SKEW PATTERNS FOLLOWED BY 200, 2000 WORD
31              ;               RECORDS OF RANDOM DATA.  AFTER ALL THE
32              ;               ONLINE, WRITE ENABLED UNITS HAVE BEEN
33              ;               WRITTEN, THEY ARE ALL READ TO INSURE
34              ;               PROPER WRITTING.  THE OPERATOR THEN INTER-
35              ;               CHANGES THE TAPES AND PERFORMS ANOTHER
36              ;               READ VERIFICATION.  THIS PROCEDURE IS CON-
37              ;               TINUED UNTIL EACH TAPE HAS BEEN READ BY
38              ;               ALL THE UNITS.  AFTER EACH READ, A SUMMARY
39              ;               OF THE ACCUMULATED STATISTICS FOR EACH
40              ;               UNIT IS PRINTED.  AFTER ALL THE UNITS HAVE
41              ;               BEEN READ, A TEST COMPLETE MESSAGE IS
42              ;               PRINTED.  IF THE OPERATOR WISHES TO CON-
43              ;               TINUE THE TEST, TYPING A 'P' CHARACTER
44              ;               WILL REPEAT THE ENTIRE TEST.
45              ;
46              ;       7.3     INTERCHANGE, READ ONLY (SA 502)
47              ;
48              ;               THE READ ONLY INTERCHANGE TEST PROVIDES
49              ;               A MEANS OF TESTING TAPE UNITS WITH PRE-
50              ;               RECORDED TAPES.  THE TAPES MUST BE RECORDED
51              ;               IN THE FORMAT DESCRIBIED BY SECTION 7.2.
52              ;               THE READ OPERATION IS IDENTICAL TO
53              ;               SECTION 7.2.
54              ;       7.4     COMMAND STRING INTERPRETER (SA 504)
55              ;
56              ;               THE COMMAND STRING INTERPRETER PROVIDES
57              ;               A TROUBLE SHOOTING AID TO ISOLATE A
58              ;               FAULT.  THE OPERATOR CAN SELECT ALL POS-
59              ;               IBLE OPERATING MODES BY RESPONDING TO
60              ;               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 RE-
;                       QUESTS 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 NUM-
;                       BER 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 PRE-
;                       VIOUS 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 SUB-
;                       SYSTEM 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

```
 0005 UMTP
01              ;               PRINTED.
02              ;8.  OPERATING MODES/SWITCH SETTINGS:
03              ;               SWITCH OPTIONS
04              ;               BIT     OCTAL   BINARY  INTERPRETATION
05              ;                       VALUE   VALUE
06              ;
07              ;               2       20000   0       ENABLE PRINT ON CONSOLE
08              ;                               1       INHIBIT PRINT ON CONSOLE
09              ;
10              ;               5       02000   0       INHIBIT LINEPRINTER
11              ;                               1       ENABLE LINEPRINTER
12              ;
13              ;               7       00400   0       ENABLE PRINT PARITY ERRORS
14              ;                               1       INHIBIT PRINT PARITY ERRORS
15                      S?WPD   8
16              ;               "ESC"   THIS COMMAND GIVEN WHILE RUNNING THE
17              ;                       ENTERED COMMAND STRING WILL CAUSE THE
18              ;                       PROGRAM TO RESTART AT THE COMMAND STRING
19              ;                       ENTER PROMPT.
20              ;
21              ;9.     OPERATING PROCEDURES/OPERATOR INPUT
22              ;
23              ;       9.1     PROGRAM LOAD
24              ;
25              ;               LOAD THE PROGRAM BY USING THE BINARY LOADER.
26              ;
27              ;       9.2     STARTING ADDRESSES
28              ;
29              ;               SA      PROGRAM FUNCTION
30              ;
31              ;               500     START RELIABILITY TEST
32              ;               501     START INTERCHANGE TEST, WRITE/READ
33              ;               502     START INTERCHANGE TEST, READ ONLY
34              ;               503     START COMMAND STRING INTERPRETER
35              ;               504     DIRECT ENTRY FOR ERROR LOG RECOVERY
36              ;
37              ;       9.3     PROGRAM OPERATION
38              ;
39              ;               9.3.1   INITIALIZATION
40              ;
41              ;                       THE FOLLOWING MESSAGE IS PRINTED REQUESTING
42              ;                       THE SETTING OF THE SOFT SWITCH REGISTER.
43              ;
44              ;       "SET SWITCH REGISTER TO DESIRED VALUE, TYPE CR TO CONTINUE."
45              ;       M
46              ;       0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
47              ;       1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
48              ;
49              ;       0
50              ;
51              ;                       MODIFY THE SWITCH REGISTER SETTING AS
52              ;                       DESCRIBIED IN SECTION 8.3, FOLLOWED BY
53              ;                       A CARRIAGE RETURN.  THE FOLLOWING MESSAGES
54              ;                       WILL BE PRINTED.
55              ;
56              ;                       IF  A REAL TIME CLOCK IS NOT PRESENT IN
57              ;                       THE SYSTEM, THE FOLLOWING MESSAGE WILL
58              ;                       BE PRINTED.
59              ;
60              ;       "TTO BAUD RATE = ?.  "
```

```
01              ;
02              ;                       RESPOND TO THE REQUEST BY TYPING THE
03              ;                       CORRECT CONSOLE DEVICE BAUD RATE FOR
04              ;                       I/O TIMING CALIBRATION.  IF THE RESPONSE
05              ;                       IS 110, THE FOLLOWING REQUEST MESSAGE
06              ;                       WILL BE PRINTED.
07              ;       "10 OR 11# BITS/CHAR = ?"
08              ;
09              ;                       RESPOND TO THE REQUEST BY TYPING 10 OR
10              ;                       11.
11              ;
12              ;                       A REQUEST TO CHANGE THE DEVICE CODE IS
13              ;                       MADE AS FOLLOWS:
14              ;
15              ;       "RELIABILITY TEST"
16              ;       "ENTER DEVICE CODE 22 OR 62, FOLLOWED BY CR.   "
17              ;
18              ;                       ANSWER THE REQUEST BY TYPING OCTAL 22
19              ;                       OR 62.  IF ANY OTHER DEVICE CODE IS
20              ;                       SELECTED, THE DEVICE CODE ENTRY PROMPT
21              ;                       WILL BE PRINTED AGAIN.
22              ;
23              ;       "ENTER 0 TO TEST CRC (NRZI ONLY) 1 IF NOT, FOLLOWED BY CR.
24              ;
25              ;       ANSWER 0 IF TAPE DRIVE IS 800 BPI NRZI OTHERWISE
26              ;       ENTER 1. NEXT A REQUEST IS MADE TO DETERMINE THE ERROR RECOVERY
27              ;       SEQUENCE THATIS TO BE USED. THIS IS DETERMINED BY THE TYPE OF OPERATING
28              ;       SYSTEM THE CONTROLLER WILL BE USED IN. THE REQUST IS MADE AS FOLLOWS:
29              ;
30              ;    "ENTER 1 IF CONTROLLER WILL BE RUN IN AOS SYSTEM 0 IF NOT,FOLLOWED BY CR"
31              ;
32              ;               9.3.2   PROGRAM ENTRY
33              ;
34              ;                       WHEN ENTERING THE RELIABILITY PROGRAM,
35              ;                       THE FOLLOWING MESSAGE WILL BE PRINTED
36              ;
37              ;       "MOUNT SCRATCH TAPE(S), ENTER CR.  TO CONTINUE."
38              ;
39              ;                       THE OPERATOR SHOULD MAKE READY ALL TAPE
40              ;                       UNITS TO BE TESTED.  ANY TAPE UNIT THAT
41              ;                       IS ONLINE WILL BE TESTED.  AFTER ALL
42              ;                       UNITS ARE READY, ENTER CR.  ON THE CON-
43              ;                       SOLE TO CONTINUE.
44              ;
45              ;               9.3.3   INTERCHANGE TEST, WRITE/READ
46              ;
47              ;                       ENTRY TO THE INTERCHANGE TEST IS IDENT-
48              ;                       ICAL TO THE RELIABILITY TEST WITH THE
49              ;                       FOLLOWING EXCEPTION.
50              ;
51              ;       "INTERCHANGE TEST(WRITE/READ)"
52              ;                       AFTER THE INITIALIZATION SECTION, THE
53              ;                       FOLLOWING MESSAGE IS PRINTED.
54              ;
55              ;       "MOUNT SCRATCH TAPE(S), ENTER CR.  TO CONTINUE."
56              ;
57              ;                       MAKE READY ALL TAPE UNITS TO BE TESTED
58              ;                       AND ENTER CR.  TO CONTINUE.
59              ;               9.3.4   INTERCHANGE TEST, READ ONLY
60              ;
```

```
;                       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"
```

```
01              ;
02              ;                       NOTE:   THE MAXIMUM WORD COUNT VALUE
03              ;                               INDICATES THE LARGEST DATA
04              ;                               BUFFER AVAILABLE.
05              ;
06              ;                       THE SUB-SYSTEM DEFAULT VALUES ARE SET
07              ;                       AS FOLLOWS:
08              ;
09              ;                       UNIT    0
10              ;                       WC      SET TO MAXIMUM WORD COUNT
11              ;                       DATA    RANDOM PATTERN
12              ;                       PARITY  ODD
13              ;
14              ;                       WHEN THE "UNIT" PROMPT IS TYPED, REFER
15              ;                       TO SECTION 7.4, FOR PROGRAM OPERATION.
16              ;10.    PROGRAM OUTPUT/ERROR DESCRIPTION
17              ;
18              ;       ALL ERRORS ARE IDENTIFIED, COUNTED AND PRINTED ON
19              ;       THE BASIS OF THE SETTING OF LOCATION "SWREG".
20              ;
21              ;       IF A UNIT GOES NOT READY, AN APPROPRIATE ERROR
22              ;       MESSAGE AND ITS ACCUMULATED STATISTICAL HISTORY
23              ;       IS PRINTED.  IF ONLY ONE(1) UNIT IS BEING TESTED,
24              ;       AN APPROPRIATE MEESAGE WILL BE PRINTED AND THE
25              ;       PROGRAM WILL WAIT FOR OPERATOR INTERVENTION.  IF
26              ;       MORE THAN ONE UNIT IS AVAILABLE, THE TEST PROCESS
27              ;       WILL CONTINUE.
28              ;
29              ;       ALL ERRORS ARE SOFT UNLESS SPECIFIED AS HARD OR
30              ;       FATAL.
31              ;
32              ;       10.1    STATISTICAL HISTORY PRINTOUT
33              ;
34              ;               THE STATISTICAL HISTORY IS PRINTED FOR
35              ;               EACH UNIT WHEN IT REACHES ITS EOT SENSOR.
36              ;               THE STATISTICAL HISTORY FOR ALL TESTED
37              ;               UNITS CAN BE REQUESTED BY TYPING A
38              ;               "SPACE" CHARACTER.  A SAMPLE OF THE
39              ;               PRINTOUT IS AS FOLLOWS:
40              ;
41              ;       "UNIT    0       1"
42              ;       "PAR WR  1       0"
43              ;       "PAR RD  1       1"
44              ;       "PERM WR 1       0"
45              ;       "PERM RD 0       0"
46              ;       "WDS RD  30348   1075827"
47              ;       "WDS WR  31345   1075827"
48              ;       10.2    STATUS WORD
49              ;
50              ;               BIT     DESCRIPTION
51              ;
52              ;               0       ANY ERROR, SET BY BITS 1,3,5,6,7,8,10,14
53              ;
54              ;               1(E)    DATA LATE
55              ;               2       REWINDING
56              ;               3(E)    ILLEGAL COMMAND
57              ;
58              ;               4       HIGH DENSITY
59              ;               5(E)    PARITY ERROR
60              ;               6(E)    EOT MARK SENSED
```

```
 0009 UNTR
01              ;
02              ;               7(E)    EOF MARK SENSED
03              ;               8(E)    BOT MARK SENSED
04              ;               9       9 TRACK TAPE
05              ;
06              ;               10(E)   BAD TAPE
07              ;               11      SEND CLOCK OR ID STATUS
08              ;               12      FIRST CHARACTER OR CORRECTED ERROR
09              ;
10              ;               13      WRITE LOCKOUT
11              ;               14(E)   CRC ERROR OR ODD REC READ
12              ;               15      UNIT READY
13                      0?DTD 11
14              ;12.    SPECIAL NOTES
15              ;
16              ;       12.1    MEDIA SELECTION
17              ;
18              ;               IT IS IMPORTANT TO SELECT KNOWN GOOD TAPES
19              ;               WHEN PERFORMING THE RELIABILITY TESTS. USING
20              ;               MARGINAL TAPE MEDIA WILL CAUSE SOFT AND HARD
21              ;               ERRORS TO OCCURE. TO VERIFIY THE SUB-SYSTEM
22              ;               RELIABILITY THE TAPE MEDIA SHOULD NOT INFLUENCE
23              ;               THE PASS OR FAIL CRITERIA.
24              ;
25              ;       12.2    DATA ENTRY
26              ;
27              ;               ALL NUMBER ENTRIES MUST BE ON OCTAL. ANY OTHER
28              ;               ENTRY WILL BE CONSIDERED AS AN ALPHA CHARACTER.
29              ;
30              ;13.    RUN TIME
31              ;
32              ;       THE PROGRAM RUN TIME IS DEPENDENT ON THE LENGTH OF THE
33              ;       TAPE MEDIA.
34                      .EOT
```

```
 0010 UNTR
**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```