

MIGHTY MUX
DIAGNOSTICS MANUAL

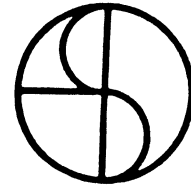
POINT 
DATA CORPORATION



E D U C A T I O N A L D A T A S Y S T E M S

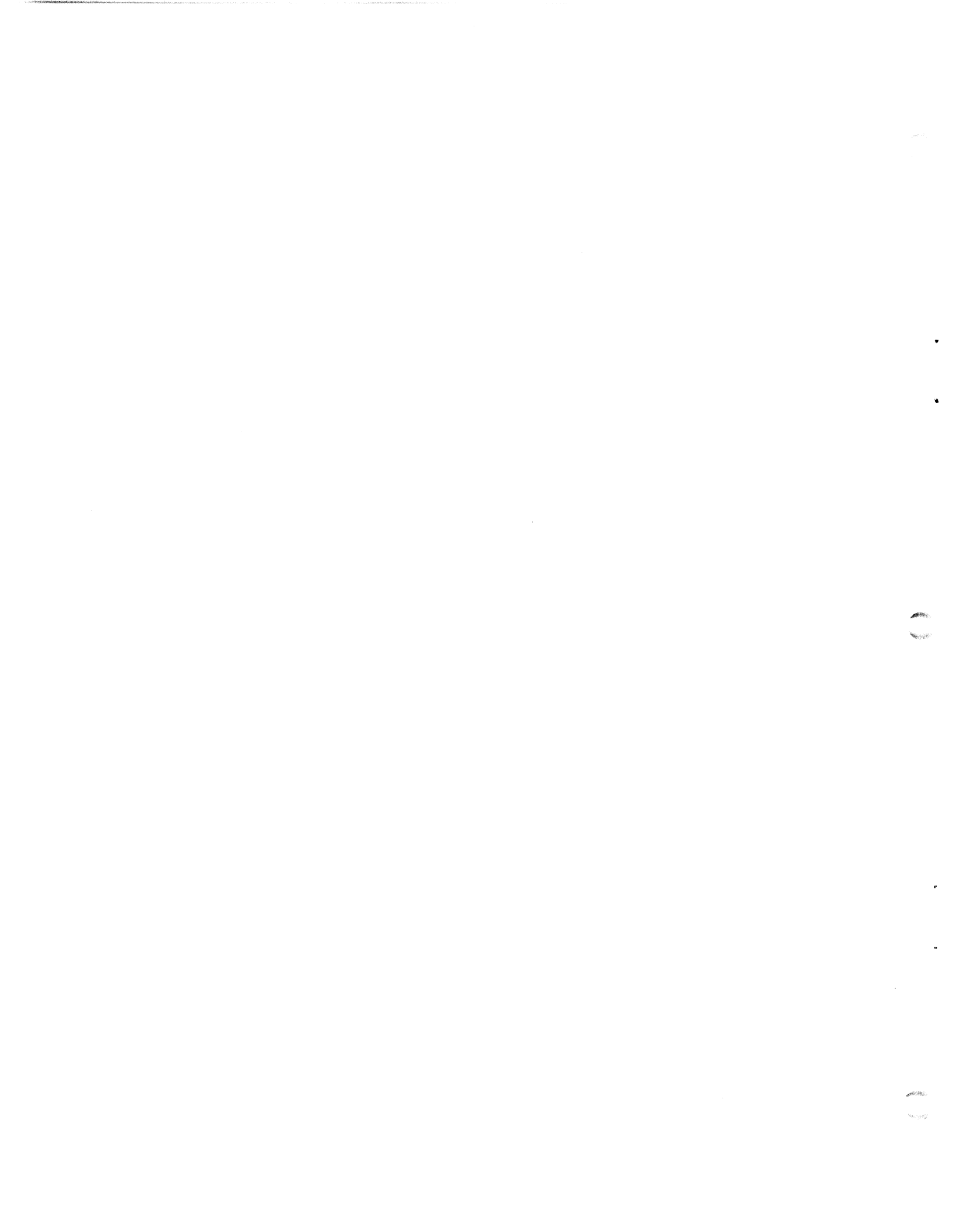
1682 Langley Avenue, Irvine, California 92714

(714) 556-4242



EDSI MIGHTY-MUX

DIAGNOSTICS



CONTENTS

EDSI MIGHTY-MUX Diagnostic Program Instruction Manual

I. Overall Description of Program	1
II. Equipment Required	1
III. Operating Procedure	2
IV. Additional Information	4
Appendix A - TTY Commands	5
Appendix B - MUXDP Flowchart	7
Appendix C - Fault Messages	9
Appendix D - How to Test the Mux With a Terminal	11
Appendix E - How to Test the MIGHTY-MUX Without a Separate I/O Board	13
Appendix F - EDSI MIGHTY-MUX Diagnostic Program Listing	15



EDSI MIGHTY-MUX Diagnostic Program

Instruction Manual

See also "EDSI MIGHTY-MUX PROGRAMMABLE DMA MULTIPLEXER
USER'S MANUAL", and MUXDP Program Listing

I. Overall Description of Program

The MIGHTY-MUX Diagnostic Program consists of two parts:

Part 1 sizes core, checks operation of the MIGHTY-MUX Real-Time clock, determines CPU speed, and tests the overall operation of the MIGHTY-MUX. At this time it determines the base address of the Mux control block area, the control block size, and the number of Mux ports operating. It types out these findings and then stops, after requesting an input from the operator to run Part 2.

Part 2 tests one port pair at a time, and runs about 200 individual tests for each port pair, testing all operational modes the Mux is capable of, such as Baud rate, character length, parity option, buffer modes, etc. A port pair consists of a transmitting port (XMTR) and a receiving port (RCVR). The output of XMTR must be connected to the input of RCVR before the test is run.

If a malfunction is found at any time, an appropriate fault message is typed out and further testing is stopped until directed by the operator.

II. Equipment Required

1. Computer - an EDSI POINT 4 Computer or any Nova-type* computer with at least 8K of core and a master terminal I/O interface board. (See Appendix D for testing MIGHTY-MUX without a separate I/O board.)
2. EDSI MIGHTY-MUX Multiplexer, with external ± 12 volt supply (any number of ports) or with inboard power option (max. 8 ports).
3. Master terminal, operating at 110 Baud, 1200 Baud, or faster. (For different Baud rate see Appendix B under "FAULT: REAL-TIME CLOCK WRONG.")

*Note: "Nova" is a Trade Mark of Data General Corporation.

III. Operating Procedure

1. Load the program into core.
2. Insure that the MIGHTY-MUX is properly installed (see "EDSI MIGHTY-MUX PROGRAMMABLE DMA MULTIPLEXER USER'S MANUAL", Appendix D). Specifically, make sure of the following:
 - a. The TTY option is disabled (see Appendix D for testing with the TTY option enabled).
 - b. MIGHTY-MUX board(s) is (are) firmly plugged in and well seated.
 - c. If CPU is not a POINT 4, it is necessary to ensure that the Data Channel Priority and Interrupt Priority signals on the computer back plane are getting to the EDSI-310 board. If the CPU has a "High-Speed" data channel, the Data Channel Priority must be jumpered so that the MIGHTY-MUX is on the standard speed data channel (on POINT 4 this is not necessary).
 - d. ±12 volt supply is plugged in and its outputs are getting from power supply to Molex-type pins on back of "asynchronous interface cable connector" (P/N 322 or 322CL) via 5-wire multicolor flat cable; and from there (via ribbon cable(s)) to the MIGHTY-MUX board(s). The ribbon cable(s) may be plugged in either way - the only effect is to reverse the port numbering (see paragraph 6 below). This step is inapplicable with onboard power option.
3. Ensure that the master terminal is turned on and connected to the computer.
4. Start the program at location 2.
5. The master terminal should now type out the following data:

Top word of core used by the diagnostic program
Effective memory cycle time
Default value of base address of Mux control area
Mux Control Block Size
Number of Mux Ports

NOTE

All these data (except memory cycle time) are given in octal form. If only the first two items are typed and then the CPU hangs with the DCH light (if any) on, it suggests that the Data Channel Priority signal is not getting to the Mux.

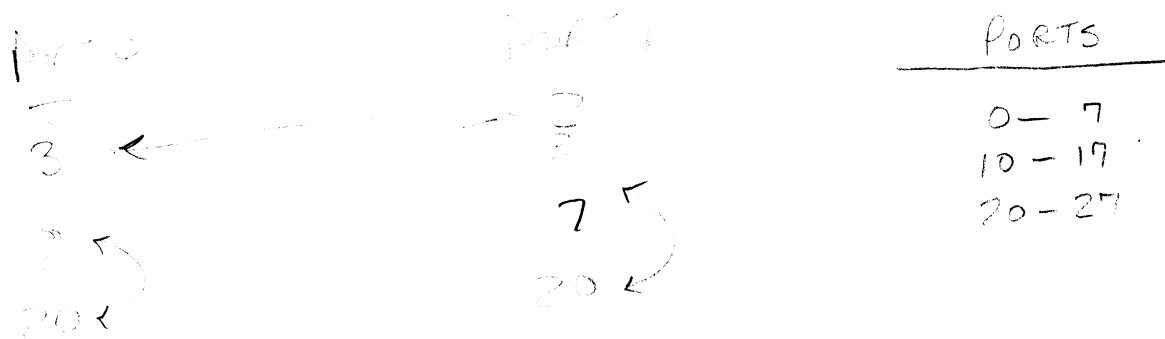
If no faults are found, the program will then type out "ENTER COMMAND", and wait for a response.

6. Connect the output of Port 0 to the input of Port 1. To do this, connect Pin 3 (of the 25-pin connector) of Port 0 to Pin 2 of Port 1, also Pin 8 of Port 0 to Pin 20 of Port 0, and Pin 7 of Port 1 to Pin 20 of Port 1. The latter two connections feed the "Device Control" output of Port 0 into its own "Device Status" input, and ground the "Device Status" input of Port 1.

NOTE

For purposes of this diagnostic program, ports are numbered octally and beginning at 0. Thus an 8-port system is numbered 0 through 7; the second 8 ports would be 10 through 17. Also note that the connector numbering on the junction panel is left-to-right when (a) the panel is oriented so that the 50-conductor ribbon cable comes off its top side, and (b) the cable is plugged into the connector on the EDSI-310 board in such a way that the cable comes out at the top. Plugging the cable in the other way round merely reverses the port numbering.

7. Type "S" (for Start). The program will now perform a sequence of about 200 individual tests. If all tests are correctly passed, it will type "AOK" after about 30 seconds and then await your next command. (See Section IV below if a fault message is typed out.)
8. Remove the connection from Port 0 to Port 1, and instead connect Port 1 to Port 2 in the same manner. Then type "N" (for Next) to start the detailed test sequence for this port pair. Proceed in this way to test all port pairs, each time moving the connection to the next port pair, typing "N", and waiting for "AOK". (Last Port gets connected to Port 0.) After the last "AOK", the diagnostic procedure is completed.



PORTS	
0 -	7
10 -	17
20 -	27

SEE APPENDIX A
PAGE A-1

IV. Additional Information

1. A selected port pair may be retested (instead of going on to the next port pair) by typing "S" (for Same). If at any time it is desired to test a port pair other than the one next in line, type in the desired XMTR port number, a comma, the desired RCVR port number, and a carriage return. The program will then type out the corresponding port addresses (core addresses where the port's control words are located). Then connect the selected XMTR to the selected RCVR and type "S" to start the full test. Any desired port may thus be tested in the role of XMTR or RCVR, with any other as its partner. Two separate ports must always be selected, however, as they are exercised in different modes in some subtests.
2. If desired, a selected port pair may be tested repeatedly by typing "F" (for Forever). This serves as a reliability test and to detect intermittent failures. The program runs the test sequence over and over until a fault occurs or "Escape" is pressed. If a fault occurs, the appropriate fault message is typed out and further testing stops.
3. If any fault messages are typed out, they may contain clues to a procedural error, such as "No -12 volt supply". Also see Appendix C. If this is no help, call EDSI at 714-556-4242.
4. If this diagnostic procedure shows the Mux to be good, but there are indications of a malfunction with a certain terminal or on a certain port, then the program may also be used to test the Mux with a live terminal. See Appendix D for how to do this.

Appendix A - TTY Commands

MUXDP contains a number of diagnostic features for the purpose of detecting and isolating possible faults. These features are controlled from the master terminal (TTY), and are summarized below.

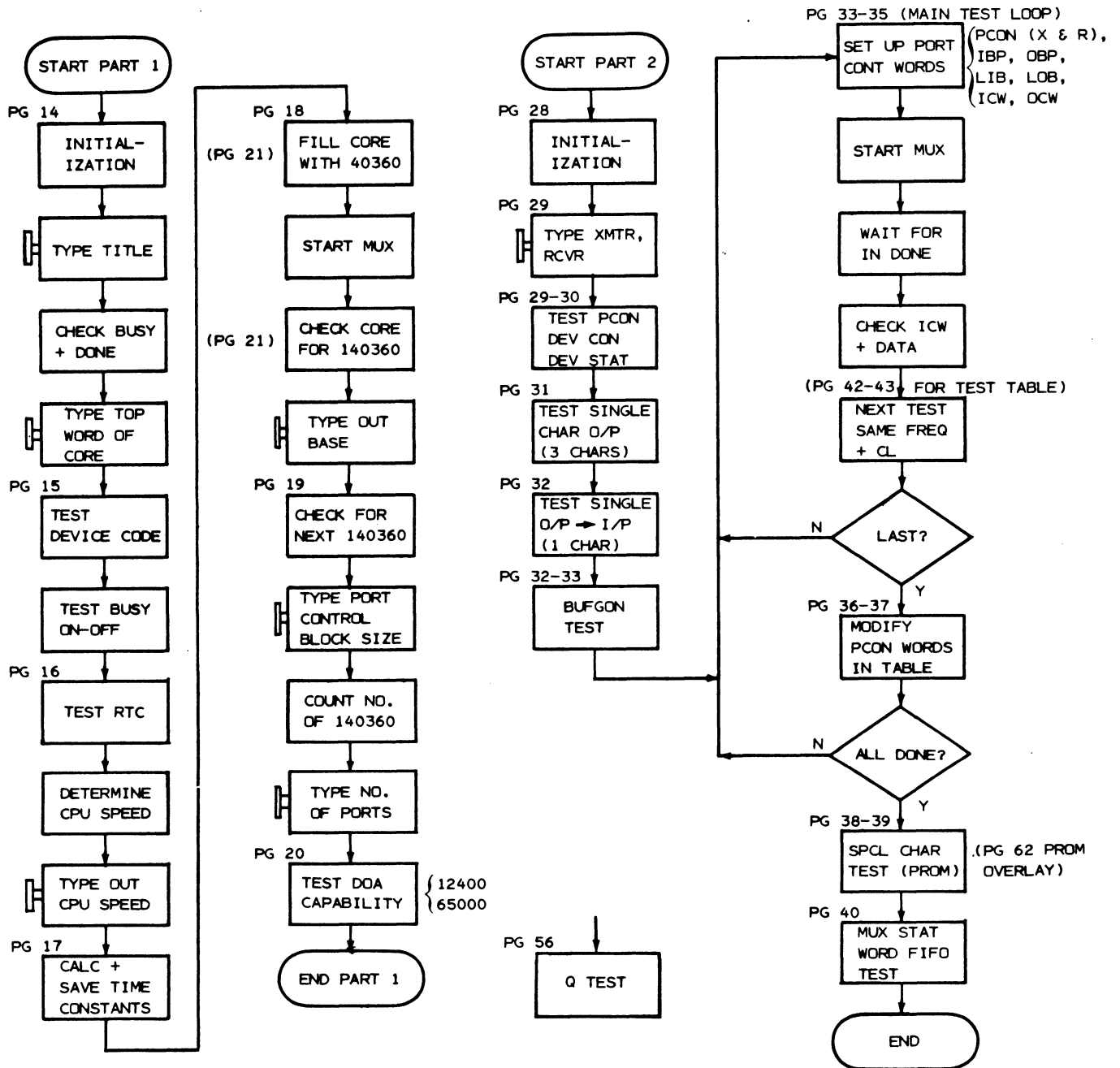
<u>TTY Entry</u>	<u>Meaning</u>
Octal Number	Specifies transmitting port (when followed by a comma) or receiving port (when followed by a carriage return). If < 200 , it is the port number (starting at 0). If ≥ 10000 , it is taken as the port control block base address.
Comma	Selects transmitting port (XMTR), if preceded by a valid octal number. Otherwise leaves XMTR as it was.
Carriage Return	Selects receiving port (RCVR), if preceded by a valid octal number (otherwise leaves RCVR as it was). Also causes type-out of XMTR and RCVR base addresses.
B	Begin Part 1 of the diagnostic over again.
C	Continue with next subtest (after a fault). If no other faults are encountered or if no more subtests are left for this port pair, types "AOK" and carriage return). Otherwise types next fault message.
D	Dump key parameters (after a fault). Types up to four lines of data: <ol style="list-style-type: none">Values in accumulators at time the fault occurred, and contents of four temporary storage cells, preceded by their addresses. (See Appendix C.)XMTR base address and its 8 control words.RCVR base address and its 8 control words.Address and contents of last automatic input buffer used.
E	Echo Test. Auto-echoes all input from RCVR until 8 characters have been received, then auto-outputs the same characters (except control characters).
F	Run selected test sequence Forever or until a fault occurs. If a fault occurs, types fault message and stops further testing.
H	Help. Types a message referencing this manual, the program's listing (including date), and EDSI's phone number.

- I IPL (Initial Program Load) to get back to normal operating software. Check program listing to determine if IPL bootstrap is appropriate for disk system you are using. Any other bootstrap may be patched in if desired.
- M Mail. Sends TTY input to XMTR, and RCVR input to TTY. If the "M" is preceded by an octal number, that octal word is first sent to XMTR as an OCW.
- N Run test sequence for Next port pair; i.e., set $XMTR = XMTR + 1$, and $RCVR = RCVR + 1$ (where Last Port + 1 = Port 0).
- Q Run a composite test which operates both the Mux and disc simultaneously to check for any data channel interference between them. Requires BZUP (for disc driver) at location 5000.
- R Repeat last subtest (which produced fault) over and over, for oscilloscope analysis.
- S Start complete test sequence for selected port pair.
- U Jump to "DEBUG" (EDSI's general-purpose debug program), if it has been loaded at 6000.
- X Jump to a short special test program for oscilloscope analysis which consists of reading the computer front panel switches and continuously entering that word into OCW of XMTR.
- Z Jump to a short special test program for oscilloscope analysis which consists of exercising all CPU I/O data and control signals (see program listing).
- (Enter Comment mode. Merely echoes all characters (on the master terminal). Terminated by an escape or a ")".
- Ctrl P (Echoes a %). Disable program's self-test feature, which verifies certain prestored checksums. Useful when making patches to program.
- Space Bar Inhibit type-out except for fault messages. If used in D (Dump) only stops current line but then types next.
- Escape Stop whatever is running, cancel previous command, and wait for next TTY command.

If an illegal command is entered, the program will type a back-slash and carriage return, and wait for the next TTY input.

A command may be entered at any time, and will terminate any type-out in process and supersede previous commands.

Appendix B - MUXDP Flowchart



(intentionally blank)

Appendix C - Fault Messages

All Fault messages are preceded by a Bell, and the message

FAULT:

appears on the screen, followed by a description of the fault and sometimes a suggestion of what might be the cause.

More diagnostic data can be obtained by typing "D", which causes the program to type out the accumulators (A0 through A3), four temporary storage cells (RTNAD, PNTR, COUNT, and .FBAK), and the XMTR and RCVR Mux control words and the last automatic input buffer used.

A3 always contains the location where the fault occurred. If this is in a subroutine, RTNAD (Return Address) contains the location from which the subroutine was called. .FBAK (Fall-back) contains the location from which the program will repeat if "R" is typed. If the fault resulted from a comparison of two values, A0 contains the expected reference value, and A1 contains the actual value which produced the fault.

To further analyze the diagnostic data, refer to the program listing. If there is any question about the correspondence of your program and listing, type H to get the release date of the program and compare that with the release date on the listing.

The remainder of this Appendix describes the meanings of certain Fault messages, together with possible causes.

FAULT: PANIC!! DIAGNOSTIC PROGRAM BEING CLOBBERED

This indicates that the program has failed its own checksum test. Possible explanations: A word in memory was changed manually, or the program was not read in correctly, or memory is defective.

A0 = stored value of checksum
A1 = computed value of checksum
A2 = location of stored value

FAULT: GARBAGE COMING IN

Some word in memory (not part of the program) has changed. |
Possible explanation: Some terminal connected to the Mux is |
producing input, or memory may be defective.

A0 = expected value of changed word
A1 = current value of changed word
A2 = address of changed word

FAULT: NO MUX ACTION -- PLUGGED IN ALL THE WAY ?

The Mux has not posted the "Output Done" bit to a valid OCW.
Possible cause: EDSI-310 board not pushed in all the way.

FAULT: REAL-TIME CLOCK WRONG

The Real-Time Clock on the Mux is tested by comparing it with the |
master terminal (TTY). This Fault message is typed out if the |
number of interrupts produced by the Real-Time Clock during the |
period between two successive TTY interrupts is not equal to 10, to |
11, or to 1. If your master terminal cannot be set to 110 Baud, or |
to a Baud rate of 1200 or higher, you may simply type C (for |
Continue) after this Fault message and continue with the test. |
Alternatively, you may change the number of expected Real-Time |
interrupts by changing the value in program location "N.RTC" (see |
last page of listing).

A0 = 12 (= 10 decimal)
A1 = number of interrupts from Real-Time Clock during TTY
interval

Appendix D - How to Test the Mux With a Terminal

The MIGHTY-MUX Diagnostic Program has two provisions - M and E - for testing the Mux with a live terminal (as opposed to testing two Mux ports against each other with a jumper between them but no terminal used). For either one, plug the test terminal into any Mux port. Tell the program which port was selected by typing in the selected port number (octal, starting at 0), followed by a comma.

It is next necessary to send the proper Port Control word (PCON) to the selected port. PCON determines the polarity of the Device Control line, number of Stop Bits, Parity Mode, Character Length, and Baud Rate. It is a 5-digit octal word having the following format.

1st octal digit = 4 for Negative Device Control Polarity
(MSD) = 5 for Positive Device Control Polarity

2nd octal digit = 0

3rd octal digit = 0 for 1 Stop Bit, Parity enabled
= 1 for 2 Stop Bits, Parity enabled
= 2 for 1 Stop Bit, Parity disabled
= 3 for 2 Stop Bits, Parity disabled

4th octal digit = 0 for 5 bit character length
= 2 for 6 bit character length
= 4 for 7 bit character length
= 6 for 8 bit character length
Add 1 for Even Parity (if enabled) \ / Odd Parity (if enabled)

5th octal digit = 0 for 110 Baud
(LSD) = 1 for 150 Baud
= 2 for 300 Baud
= 3 for 600 Baud
= 4 for 1200 Baud
= 5 for 2400 Baud
= 6 for 4800 Baud
= 7 for 9600 Baud

0,50277E

Determine the required PCON and type it in on the master terminal (TTY), followed by an M or E. This transmits the PCON to the selected port and sets it in "Mail" or "Echo" mode. To reenter M or E after escaping from them, it is sufficient to type M or E if Port Number and PCON have not been changed in the meantime.

- M Mail. Anything typed on the master terminal (until escape) will be sent as output to the Mux terminal, and any input received from the Mux terminal will be typed out on the master terminal.
- E Echo Test. Sets up an 8-byte automatic input buffer and puts RCVR into auto-echo mode. When the buffer is full (or if a parity error occurs), starts auto output from the same buffer, using the Special Interrupt Request feature. Thus, anything typed on the terminal will be echoed to it, until 8 characters have been echoed. At that time the same 8 characters will be retransmitted to the terminal, up to (and including) the first control character (if any).

Appendix E - How to Test the MIGHTY-MUX Without a Separate I/O Board

If your Mux has the TTY option enabled, it is possible to run the Mux Diagnostic without a separate I/O controller in the CPU, with a few modifications.

The Master TTY option on the MIGHTY-MUX is enabled by means of jumper wire connections. Depending on the revision number of the 310 board, the option will be enabled as follows:

- 1) Rev. A & B: Enabled by means of a jumper wire near location 3C, pin 8 (etch near location 3C, pin 8 must be cut). When a jumper is installed over the letter "T," the TTY option is enabled.
- 2) Rev. C or Higher: Enabled by means of two jumper wires:
 - a) a jumper wire installed over the letter "T" near location 3C, pin 8 (etch near location 3C, pin 8 must be cut); and
 - b) a jumper wire installed over the letter "T" between locations 7C and 8C.

When the TTY option is enabled, the Mux can be used to communicate with a terminal in the same way that a standard I/O board does. As long as the multiplexing function is not turned on, Port 0 of the Mux will respond to standard TTY software, i.e., device code 10/11 type I/O commands. Once the multiplexing function has been turned on (by means of a DOC -,MUX instruction), Port 0 will respond to the Mux software in the same manner as all other ports.

When running MUXDP on a Mux with the TTY option enabled, the following points must be observed.

1. The master terminal is of course connected to Port 0 on the Mux junction panel.
2. Part 1 of the diagnostic will run precisely as with a separate I/O board.
3. Part 2 will run normally with all port pairs if Port 0 is not involved (since Port 0 is used for the master terminal).
4. To run Part 2 with Port 0 as XMTR or RCVR, start the test as you normally would (i.e., type S or N). The program will type out a Fault message, because Port 0 is still connected to the master terminal and not to the other port. Type T. The computer will halt. Connect the ports as required, disconnecting the master terminal. When finished, press CONTINUE on the computer front panel. The test will then run as usual for about 30 seconds. After that time reconnect the master terminal and then press R to see if the test ended with "AOK" or with a fault. If "AOK" is typed the test was passed.

5. If it is desired to observe the diagnostic messages as they are typed out, this can be accomplished at the expense of constructing a special test cable. Simply connect the output of Port 0 to the master terminal in parallel with any other connection required. (This can not be done for the input connections because inputs cannot be paralleled.)
6. If it is desired to run the master terminal at a Baud rate other than 110 Baud, determine the proper PCON word for the desired Baud rate (see Appendix D for how to construct the PCON word), and patch it into the diagnostic at location MRATE (MUXDP listing page 4).

Appendix F - EDSI MIGHTY-MUX
Diagnostic Program Listing

ASM ,@*LPT,MUXDP.P4.N
AUG 6, 1979 11:43:25

; MUXDP -- DIAGNOSTIC PROGRAM FOR EDSI MIGHTY-MUX MULTIPLEXER (MODEL 310)
; 06-27-79

; ALL RIGHTS RESERVED
; COPYRIGHT (C) 1976, EDUCATIONAL DATA SYSTEMS, INC.
; 1682 LANGLEY AVE., IRVINE, CALIFORNIA 92714

; THIS DIAGNOSTIC PROGRAM TESTS THE MIGHTY-MUX MULTIPLEXER (SEE DOCUMENT
; EDS-1028). THE PROGRAM STARTS AT 2 AND CONVERSES WITH THE
; OPERATOR VIA TELETYPE. TO SKIP PART 1 OF THE PROGRAM START AT 3.

; PART 1 DETERMINES THE EQUIPMENT CONFIGURATION (CORE SIZE,
; CPU SPEED, MUX BASE ADDRESS, CONTROL BLOCK LENGTH, AND NUMBER OF
; PORTS). PART 2 PERFORMS ABOUT 200 SEPARATE SUBTESTS WHICH TEST ALL
; CAPABILITIES OF THE MUX. PART 2 REQUIRES THAT A TRANSMITTING PORT
; (XMTR) AND A RECEIVING PORT (RCVR) BE SELECTED AND INTERCONNECTED.
; CONNECT XMTR PIN 3 TO RCVR PIN 2 (DATA)
; AND XMTR PIN 8 TO XMTR PIN 20 (CONTROL ==> STATUS)
; AND RCVR PIN 7 TO RCVR PIN 20 (GROUND STATUS LINE)

; SUMMARY OF COMMANDS (SEE EDS-1028)

; OCTAL # XMTR OR RCVR, EITHER PORT NO. OR BASE ADDRESS
; , SELECT PREVIOUS OCTAL NO. AS XMTR
; CR SELECT PREVIOUS OCTAL NO. AS RCVR, & TYPE OUT XMTR AND RCVR
; B BEGIN ALL OVER AGAIN (PART 1)
; C CONTINUE WITH NEXT SUBTEST
; D DUMP A0, A1, A2, A3, RTNAD, PNTR, COUNT, .FBAK; CONTROL WORDS
; E ECHO: SETS XMTR IN AUTO ECHO MODE (FOR USE WITH A TERMINAL).
; F RUN SELECTED TEST SEQUENCE FOREVER
; H HELP
; I IPL BACK TO IRIS (IF AVAILABLE)
; M MAIL: SEND TTY INPUT TO XMTR, AND RCVR INPUT TO TTY.
; N SELECT NEXT PORT PAIR AND START PART 2
; Q COMPOSITE TEST EXERCISING BOTH DISC AND MUX (REQ. BZUP AT 5000)
; R REPEAT FAILED SUBTEST FOR SCOPE ANALYSIS
; S START COMPLETE TEST (PART 2) FOR SELECTED PORT PAIR
; T TEST WITH TTY OPTION: HALTS, CHANGE HOOKUP, PRESS CONTINUE
; U GO TO DEBUG, IF LOADED
; X SHORT SPECIAL TEST (READ SWITCHES INTO DCW)
; Y* TEST SYNCHRONOUS PORT (EDS-302 BOARD)
; Z EXERCISE CPU I/O CONTROL SIGNALS
; CTRL P DISABLE CORE CHECKSUM VERIFICATION
; (ENTER COMMENT MODE (ECHOES ALL CHARS. UNTIL ")" OR ESCAPE)
; ESC CANCEL PREVIOUS COMMAND
; SPACE INHIBIT TYPE-OUT EXCEPT FAULTS
; * NOT PART OF BASIC DIAGNOSTIC PROGRAM

; TABLE OF CONTENTS
; -----
; PAGE ZERO (CONSTANTS, VARIABLES, SUBROUTINES) 2-13
; PART 1 OF DIAGNOSTIC (SIZE CORE, ETC.) 14-21
; WAIT FOR TTY INPUT AND RESPOND TO IT 22-27
; PART 2 OF DIAGNOSTIC (DETAILED TESTS) 28-43
; FAULT AND ERROR PROCESSING ROUTINES 44-45
; MESSAGES FILE 46-55
; "Q" - COMPOSITE DISC/MUX TEST 56-61

	0	.LOC	0	
0	0	0		
1	1163	INTF		; INITIAL INTPT. SERVICE ADDRESS
2	2004	JMP	@.STRT	; START PART 1 OF TEST
3	2006	JMP	@.WAIT	; ENTER COMMAND MODE

; DEFINITIONS

10	.DUSR	TTI=	10
11	.DUSR	TTO=	11
25	MUX=	25	
2401	GO=	2401	
400	L=	400	

; MUX CONTROL WORD DISPLACEMENTS

0	ICW. =	0
1	OCW. =	1
4	IBP. =	4
5	OBP. =	5
6	LIB. =	6
7	LOB. =	7

; CONTROL CODES FOR CONDITION LISTS (SEE "IOCK")

0	ICW. T=	ICW. *4	
1	ICW. R=	ICW. *4+1	
4	OCW. T=	OCW. *4	
5	OCW. R=	OCW. *4+1	
2	MSK=	2	; MASKS OUT STATUS BITS BEFORE COMPARING
21	IBP. R=	IBP. *4+1	
24	OBP. T=	OBP. *4	
100004	MSW. =	100004	; MUX STATUS WORD - TEST ONLY BITS 0 & 1
100000	MSW. T=	100000	; INSERT XMTR PORT ADDRESS
100001	MSW. R=	100001	; INSERT RCVR PORT ADDRESS
2	S=	2	; GIVE S PULSE AFTER READING MSW

; POINTERS

4	675	.STRT:	START	
5	711	.ST2:	STRT2	
6	1402	.WAIT:	WAIT	
7	1410	.WT1:	WAIT1	
10	1400	.ABDR:	ABORT	
	6011	TEST=	JSR	@.
11	3374		TEST.	
12	3376	.FALT:	FAULT	
13	623	.ENTP:	ENBTP	; ENABLE TYPE-OUT
14	2733	.S960:	SET96	; SET XMTR AND RCVR TO 9600 BAUD
15	153716		153716	; CHECKSUM OVER WORDS 2-14
16	0		0	; RESERVED FOR BREAKPOINTS
17	0		0	

20 77377 T1MS: VAR. ; THESE VALUES ARE FILLED IN BY PGM
21 77377 VAR. ; AFTER DETERMINING CPU SPEED
22 77377 T2MS: VAR. ; (SEE "SPEED")
23 77377 VAR.
24 77377 T10MS: VAR.
25 77377 VAR.
26 77377 T50MS: VAR.
27 77377 VAR.
30 77377 T. 1SE: VAR.
31 77377 VAR.
32 77377 T. 2SE: VAR.
33 77377 VAR.
34 100000 T. 4SE: 100000 ; INITIAL VALUE ONLY
35 100000 100000
36 0 T. 5SE: 0 ; THIS IS ALWAYS 0 (FOR MAX TIME)
37 0 0

77377 VAR. = 77377 ; PREVENTS UNNECESSARY PUNCHING OF VARIABLES

; CONSTANTS

40 3 C3: 3
41 4 C4: 4
42 5 C5: 5
43 6 C6: 6
44 7 C7: 7
45 10 C10: 10
46 12 C12: 12
47 15 C15: 15
50 20 C20: 20
51 25 CMUX: MUX
52 31 C31: 31
53 40 C40: 40
54 60 C60: 60
55 67 C67: 67
56 101 C101: 101
57 177 C177: 177
60 4000 C4000: 4000
61 6000 C6000: 6000
62 7740 C7740: 7740
63 10000 C10K: 10000
64 20000 C20K: 20000
65 40360 C40KX: 40360
66 60000 C60K: 60000
67 140000 C140K: 140000
70 170000 C170K: 170000
71 177400 C377L: 177400
72 170377 MSTAT: 170377 ; DCW MASK TO EXCLUDE STATUS BITS

73 110 . A0: A0
74 106 . TSO: TSO
75 130 . TSN: TSN
76 20 . TVAR: T1MS
77 132572 CKS0: 132572 ; CHECKSUM

; VARIABLES (NOT INCLUDED IN CORCK CHECKSUM):

100 10777 BOTWD:10777
101 77577 TOPWD:77577 ; MAY BE CHANGED TO PROTECT MORE OR LESS CORE
102 12 N.RTC:12 ; NO. R-T CLOCK INTPTS PER TTY PERIOD
103 0 MRATE:0 ; PCON TO SET MASTER TERMINAL TO PROPER RATE

104 60200 XNIOC:NIOC 0 ; DEVICE CODE WILL BE CHANGED BY PGM
105 1400 JMP 0,3 ; (SEE "DEVCO")

; VARIABLES TO BE INITIALIZED TO ZERO:

106 TSO=

106 77377 CMD: VAR. ; CMD = COMMAND WORD
; BIT 0 (MSB) = R = REPEAT FOR SCOPE ANALYSIS
; BIT 1 = F = RUN FOREVER; TYPE OUT FAULTS ONLY
; BIT 15 = INHIBIT TYPE-OUT UNTIL NEXT FAULT OR INPUT

107 77377 SPFLG:VAR. ; FLAGS WHERE TO GO WHEN "SPACE" IS PRESSED
; 0 (INITIAL VALUE) = GO @. FBAK
; 1 (AFTER PART 1) = GO TO "WAIT"
; 2 (DURING DUMP) = GO TO "CDUMP"

110 77377 A0: VAR.
111 77377 A1: VAR.
112 77377 A2: VAR.
113 77377 A3: VAR.
114 77377 RTNAD:VAR.
115 77377 PNTR: VAR.
116 77377 COUNT:VAR.
117 77377 .FBAK:VAR.

120 77377 TS: VAR. ; TEMP. STORE, MAIN PROGRAM
121 77377 VAR.
122 77377 VAR.
123 77377 VAR.
124 77377 VAR.
125 77377 TSR: VAR. ; TEMP. STORE, SUBROUTINES
126 77377 VAR.
127 77377 VAR.

130 77377 CCFLG:VAR. ; CORE CHECK FLAG, PREVENTS CORE CHECK IF > 0

; ALIASES FOR TEMPORARY STORAGE CELLS:

120 TSCPU=TS ; NO. TIMING LOOPS IN 10 MS (SEE "SPEED")
121 TSTK= TS+1 ; POINTER TO REFERENCE TIME CONSTANTS
122 TSTV= TS+2 ; POINTER TO VARIABLE TIME CONSTANTS
123 INISW=TS+3 ; INITIALIZATION SWITCH FOR MODPC (SEE "INLST")
124 SWICH=TS+4

; END OF VARIABLES TO BE INITIALIZED TO ZERO

130 TSN= -1

131 77577 CORSZ: 77577 ; CORE SIZE MINUS 200
132 36000 BASE: 36000 ; BASE ADDRESS (ICW) OF PORT 0 CONTROL BLOCK
133 40 BLKL: 40 ; MUX CONTROL BLOCK LENGTH
134 36340 LPORT: 36340 ; BASE ADDRESS OF LAST PORT CONTROL BLOCK
135 0 DELTA: 0 ; AMOUNT BY WHICH IBP IS INCREASED EACH TEST
136 36040 XMTR: 36040 ; BASE ADDRESS OF CURRENTLY USED XMTG PORT
137 36100 RCVR: 36100 ; BASE ADDRESS OF CURRENTLY USED RCVG PORT
140 21777 IBP: 21777 ; INPUT BYTE POINTER (MOVES THROUGH CORE)
141 11000 BFR: 11000 ; WORD ADDRESS CORRESPONDING TO IBP

; CONTROL WORDS FOR X-TEST

142 40377 XPCON: 40377 ; PCON USED IN X-TEST FOR RCVR
143 120252 XBFR: 240*L+252 ; AUTO OUTPUT BUFFER FOR X-TEST
144 305 XOBP: XBFR*2-1
145 307 XLOB: XBFR*2+1

; END OF VARIABLES. ALL OTHER WORDS ARE INCLUDED IN CHECKSUM

; CORE CHECK -- CHECK PROGRAM'S CHECKSUM TO INSURE ITS INTEGRITY

146 102401 CORCK: SUB 0, 0, SKP
147 20156 CORCX: LDA 0, C2 ; EXCLUDE "LIST" FROM CORCK
150 102257 SKIP
151 20204 CCIPL: LDA 0, CCIPC ; CHECK ONLY IPL SEQUENCE
152 54114 STA 3, RTNAD
153 4201 JSR CORC1

154 3253 .LIST: LIST ; START OF A CHECKSUM BLOCK
155 3373 CKS6 ; END OF CHECKSUM BLOCK
156 2 C2: 2
157 15 15
160 40 C3
161 77 CKS0
162 146 CORCK
163 266 CKS1
164 315 MSG
165 366 CKS2
166 401 C401: 401
167 1147 CKS3
170 1150 CKS3+1
171 2043 CKS4
172 2044 CKS4+1
173 3252 CKS5
174 3374 CKS6+1
175 4321 CKS7
176 367 IPL
177 400 C400: 400
200 0 0

201 117000 CDRC1: ADD 0, 3
202 2203 JMP @. CDRC

203 644 . CDRC: CDRC2
204 22 CCIPC: C400-. LIST-1

; FBAK: SAVE A FALLBACK LOCATION (FOR "REPEAT" AND "CONTINUE")

```
205 62677 FBAK: IORST
206 62677 IORST
207 62677 IORST
210 30136 LDA 2,XMTR
211 102400 SUB 0,0
212 24106 LDA 1,CMD
213 125113 SSN 1,1 ; IS CMD = REPEAT ?
214 54117 STA 3,FBAK ; NO, SAVE NEW FALLBACK
215 2117 JMP @,FBAK ; YES, RETURN TO OLD FALLBACK
```

; SUBROUTINES TO SET UP INTERRUPT SERVICE

```
216 10 CTTI: TTI
217 177775 CM3: 177775 ; MASK FOR TTY INPUT
220 3242 ISTTY ; INTERRUPT SERVICE ADDRESS
221 20217 EITTY: LDA 0,-2 ; ENABLE INTERRUPTS FROM TTI
222 24220 LDA 1,-2
223 230 JMP EINT
```

```
224 175777 175777
225 3225 ISMUX
226 20224 EIMUX: LDA 0,-2 ; ENABLE INTERRUPTS FROM MUX
227 24225 LDA 1,-2
230 60277 EINT: INTDS
231 62077 MSKD 0
232 44001 STA 1,1
233 60177 INTEN
234 1400 JMP 0,3
```

; WAIT FOR CONVENIENT OUTPUT TIME: JUST AFTER REAL-TIME INTPT.

```
235 60525 OUTIM: DIAS 0,MUX ; CLEAR MUX DONE FLAG
236 60325 NIOP MUX ; START REAL-TIME CLOCK
237 54114 STA 3,RTNAD
240 126400 SUB 1,1
241 46137 STA 1,@RCVR
242 63725 SKPDZ MUX
243 250 JMP .+5
244 125404 INC 1,1,SZR
245 242 JMP .-3
246 6012 JSR @,FALT
247 3650 FNRTC
```

```
250 20132 LDA 0,BASE
251 61025 DDA 0,MUX
252 1400 JMP 0,3
```

; STORE THE OUTPUT CONTROL WORD GIVEN JUST AFTER THE JSR CALL

```
253 62025 STOCW: DOB 0, MUX
254 21001 LDA 0, DCW. , 2 ; STORE DCW, PRESERVING STATUS BITS
255 24061 LDA 1, C6000
256 107400 AND 0, 1
257 21400 LDA 0, 0, 3
260 107001 ADD 0, 1, SKP
261 25400 STOCU: LDA 1, 0, 3 ; ENTER HERE TO OVERWRITE STATUS BITS
262 45001 STA 1, DCW. , 2
263 63025 DDC 0, MUX
264 61025 DOA 0, MUX ; PROD MUX FOR IMMEDIATE OUTPUT
265 1401 JMP 1, 3
```

```
266 44577 CKS1: 44577 ; CHECKSUM
```

; PATCH SPACE

; TYPE-OUT ROUTINES

```
; "TYPE" TYPES THE CHARACTER STRING FOLLOWING THE JSR
; "TPMSG" TYPES THE CHARACTER STRING POINTED TO BY A2
; "MSG" TYPES THE CHARACTER STRING POINTED TO BY THE WORD FOLLOWING
; THE JSR AND RETURNS TO THE NEXT WORD
; IN THE ABOVE, "CHARACTER STRING" MAY BE (INTERMIXED) EITHER 2 ASCII
; CHARACTERS/WORD (IF MSB=0) OR 3 ALPHA CHARACTERS/WORD (IF MSB=1)
; "TPOCT" TYPES THE OCTAL NUMBER IN A0 WITHOUT LEADING ZEROS OR SPACES
; "TPOCL" TYPES THE OCTAL NUMBER IN A0 WITH LEADING SPACES FOR ZEROS
; "TPDEC" CONVERTS THE NUMBER IN A1 TO BCD AND TYPES IT AS DECIMAL
; IN ALL CASES IF TTY INPUT COMES IN THE OUTPUT IS DISCONTINUED
; AND THE PROGRAM JUMPS TO "WAIT1"
```

```
26 .LOC 315- ; .LOC OVERFLOW CHECK
315 .LOC 315
```

```
315 54125 MSG: STA 3, TSR
316 20106 LDA 0, CMD
317 101015 SNZ 0, 0
320 6013 JSR @. ENTP
321 32125 LDA 2, @TSR
322 10125 ISZ TSR
323 327 JMP . +4
324 171000 TYPE: MOV 3, 2
325 176000 ADC 3, 3
326 54125 TPMSG: STA 3, TSR
327 50126 STA 2, TSR+1
330 451 JMP TPNXT
```

```
331 152521 TPOCT: SUBZL 2, 2, SKP ; "NULL" FOR NO LEADING SPACES
332 30053 TPOCL: LDA 2, C40 ; "SPACE" FOR LEADING SPACES
333 54125 STA 3, TSR
334 24064 LDA 1, C20K ; PREPARE MASK FOR ONE OCTAL DIGIT
335 44126 STA 1, TSR+1
336 127120 ADDZL 1, 1 ; CHANGE MASK FOR FIRST DIGIT = 1 BIT
337 101041 TPDIQ: MOVQ 0, 0, SKP ; PREPARE TO INSERT "PUSHER" BIT IN A0
340 24126 LDA 1, TSR+1 ; LEFT-SHIFT ONE DIGIT FROM A0 INTO A1
341 101105 MOVL 0, 0, SNR ; EXIT WHEN "PUSHER" BIT IS GONE
342 2125 JMP @TSR
343 125103 MOVL 1, 1, SNC
344 341 JMP -3
345 125025 MOVZ 1, 1, SNR ; NON-ZERO DIGIT ?
346 101135 MOVZL# 0, 0, SNR ; OR LAST DIGIT ?
347 30054 LDA 2, C60 ; YES: ADDEND FOR ASCII DIGIT
350 4442 JSR TPCHA-1
351 340 JMP TPDIQ+1

352 513 TPDEC: JMP TPDE1
```

; CONDITIONAL TIMING CHECKS (SEE IOCK)

```
353 30136 OUTCK: LDA 2, XMTR ; WAIT FOR XMTR OUT DONE BIT
354 151400 INC 2, 2
355 533 JMP IOCK

356 102400 INCK0: SUB 0, 0 ; CLEAR RCVR ICW AND ...
357 42137 STA 0, @RCVR
360 30137 INCK: LDA 2, RCVR ; WAIT FOR RCVR IN DONE BIT
361 527 INCKX: JMP IOCK ; WAIT FOR XMTR IN DONE BIT

362 126000 CHECK: ADC 1, 1 ; CHECK PARAMETERS WITHOUT WAITING
363 137001 ADD 1, 3, SKP
364 27400 TIMCK: LDA 1, @0, 3 ; WAIT A FIXED TIME
365 540 JMP TIMCH

366 130717 CKS2: 130717 ; CHECKSUM
```

```
; INITIAL PROGRAM LOAD BOOTSTRAP FOR IRIS WITH AN S. I. 3045 CONTROLLER
; READS DISC BLOCK 0 FROM FIXED DISC INTO CORE PAGE 0 AND JUMPS TO 377
; FOR ANOTHER CONTROLLER PATCH IN DESIRED BOOTSTRAP STARTING AT
; 370 AND ADJUST CHECKSUM AT 400
```

```

      0 .LDC 367-.          ;.LDC OVERFLOW CHECK
      367 .LDC 367

367  4151 IPL:  JSR  CCIPL      ;INSURE IPL CHECKSUM IS OK
370  20404      20404 ;LDA 0, .+4
371  4404      4404  ;JSR .+4
372  400        400
373  0          0
374  0          0      ;FOR CARTRIDGE, 100000 (20000 FOR 3015)
375  77140     DDCS  3, DSC
376  63240     DOCC  0, DSC
377  377 C377: 377

400  23634     23634      ;CHECKSUM FOR IPL BOOTSTRAP

      40 DSC= 40
```

```
; CONTINUATION OF TYPE-OUT ROUTINES
```

```

401  22126 TPNXT: LDA  0, @TSR+1
402  10126      ISZ  TSR+1
403  101112     SSP  0, 0
404  433        JMP  TP3CH      ;TYPE 3 PACKED ALPHA CHARACTERS
405  105300     MOVS 0, 1      ;TYPE 2 STANDARD ASCII CHARACTERS
406  4405      JSR  TPCHA
407  105000     MOV  0, 1
410  4403      JSR  TPCHA
411  770       JMP  TPNXT
```

```
; TYPE THE RIGHT BYTE IN A1 (IF NON-ZERO); PRESERVES A0, A2, AND CARRY.
; IF BYTE=0, RETURNS @TSR UNLESS TSR=-1 IN WHICH CASE IT RETURNS @TSR+1
```

```

412  147000     ADD  2, 1
413  54127 TPCHA: STA  3, TSR+2
414  60277     INTDS
415  63710     SKPDZ TTI
416  2007      JMP  @.WT1
417  63511     SKPBZ TTO
420  777       JMP  .-1
421  34057     LDA  3, C177
422  167405     AND  3, 1, SNR ; IS CHAR = TERMINATOR ?
423  410       JMP  TPRTN      ; YES - RETURN
424  125234     MOVZR# 1, 1, SZR ; IS CHAR = 1 (NULL) ?
425  34106     LDA  3, CMD      ; NO
426  175213     MOVR# 3, 3, SNC ; IS CMD = NO-TYPE OR CHAR = 1 ?
427  65111     DOAS  1, TTO      ; NO - THEN TYPE IT
430  63511     SKPBZ TTO
431  777       JMP  .-1
432  2127     JMP  @TSR+2
```

```
433 34125 TPRTN: LDA 3, TSR ; RETURN FROM TYPE ROUTINE
434 175414 INC# 3, 3, SZR
435 1400 JMP 0, 3
436 2126 JMP @TSR+1

437 101140 TP3CH: MOVOL 0, 0 ; INSERT "PUSHER" BIT
440 24060 LDA 1, C4000 ; SHIFT A 5-BIT CHAR. INTO A1
441 101125 MOVZL 0, 0, SNR
442 737 JMP TPNXT ; EXIT WHEN PUSHER BIT IS GONE
443 125103 MOVL 1, 1, SNC
444 775 JMP .-3
445 34052 LDA 3, C31
446 136032 SGE 1, 3 ; NON-ALPHA CHARACTER ?
447 413 JMP TP3C1 ; NO
450 4410 JSR .+10 ; YES - SEE MSG. FOR EXPL. OF CODING
451 177757 "0-H ; Z ==> 0
452 177737 " -H ; [ ==> SPACE
453 177677 00-H ; \ ==> END STRING
454 177754 "--H ; ] ==> -
455 177711 12-H ; ^ ==> LF
456 177714 15-H ; _ ==> CR
457 177774 "=-H ; ` ==> =
460 137000 ADD 1, 3
461 25747 LDA 1, -31, 3
462 30056 TP3C1: LDA 2, C101
463 4727 JSR TPCHA-1
464 754 JMP TP3CH+1
```

; CONVERT THE BINARY NUMBER IN A1 (MAX 9999) TO BCD AND TYPE IT
; IN DECIMAL.

```
465 54125 TPDE1: STA 3, TSR
466 102620 SUBZR 0, 0 ; WILL ACCUMULATE BCD; CONTAINS PUSHER
467 30046 TPDE2: LDA 2, C12 ; DIVIDE A1 BY 10
470 176401 SUB 3, 3, SKP
471 175400 INC 3, 3 ; A3 = QUOTIENT
472 146422 SUBZ 2, 1, SZC
473 776 JMP .-2
474 147000 ADD 2, 1 ; A1 = REMAINDER (< 10)
475 125200 MOVR 1, 1 ; SHIFT REMAINDER (BCD) INTO A0
476 101202 MOVR 0, 0, SZC
477 405 JMP TPDE3 ; WHEN PUSHER COMES OUT, CONV. IS DONE
500 151224 MOVZR 2, 2, SZR ; COUNT 4 BITS TO BE SHIFTED
501 774 JMP .-4
502 165000 MOV 3, 1 ; PREPARE TO DIVIDE QUOTIENT AGAIN
503 764 JMP TPDE2

504 152520 TPDE3: SUBZL 2, 2 ; "NULL" FOR NO LEADING SPACES
505 24063 LDA 1, C10K ; PREPARE MASK FOR ONE BCD DIGIT
506 44126 STA 1, TSR+1
507 337 JMP TPDIG
```


; TIMED CONDITION CHECK SUBROUTINES

; OUTCK WAITS FOR XMTR OUT DONE BIT = 1
; INCK WAITS FOR RCVR IN DONE BIT = 1
; INCKX WAITS FOR XMTR IN DONE BIT = 1
; IN EACH CASE THE WAIT IS SUBJECT TO A MAXIMUM AND A MINIMUM TIME
; AS SPECIFIED BY THE FIRST WORD FOLLOWING THE JSR CALL
; TIMCK WAITS THE SPECIFIED AMOUNT OF TIME
; CHECK DOESN'T WAIT AT ALL
; ALL SUBROUTINES THEN CHECK THE CONDITIONS SPECIFIED BY
; SUBSEQUENT PAIRS OF WORDS, WHERE THE FIRST WORD OF EACH PAIR
; SPECIFIES THE CONDITION (SEE "CONTROL CODES" ON PAGE 2)
; AND THE SECOND THE EXPECTED VALUE

```
510 54114 IOCK: STA 3,RTNAD
511 27400 LDA 1,@0,3
512 125140 MOVOL 1,1
513 21000 LDA 0,0,2 ; \
514 101112 SSP 0,0 ; \
515 423 JMP MINTM ; \
516 22400 LDA 0,@ ; \ STANDARD
517 22400 LDA 0,@ ; / TIMING LOOP
520 125405 INC 1,1,SNR ; /
521 101002 MOV 0,0,SZC ; /
522 771 JMP .-7 ; /
523 6012 JSR @.FALT
524 3746 FTIMO may time out

525 54114 TIMCH: STA 3,RTNAD
526 125140 MOVOL 1,1
527 21000 LDA 0,0,2 ; FILLER INSTRUCTIONS
530 102257 SKIP ; TO EQUALIZE COUNT-DOWN LOOPS
531 63077 HALT
532 22400 LDA 0,@
533 22400 LDA 0,@
534 125405 INC 1,1,SNR
535 101002 MOV 0,0,SZC
536 771 JMP .-7
537 410 JMP TCOND

540 32114 MINTM: LDA 2,@RTNAD ; CK FOR MINIMUM ELAPSED TIME
541 21001 LDA 0,1,2
542 125200 MOVR 1,1
543 122033 SLS 1,0
544 403 JMP .+3
545 6012 JSR @.FALT
546 3753 FMINT
```

547	10114	TCOND: ISZ	RTNAD	; TEST NEXT ITEM IN CONDITION LIST
550	36114	LDA	3, @RTNAD	
551	10114	ISZ	RTNAD	
552	30136	LDA	2, XMTR	
553	175415	INC#	3, 3, SNR	; IS CONDITION LIST FINISHED ?
554	2114	JMP	@RTNAD	; YES - RETURN
555	175222	MOVZR	3, 3, SZC	; NO - DOES IT REF. RCVG. PORT ?
556	30137	LDA	2, RCVR	; YES
557	22114	LDA	0, @RTNAD	
560	177112	ADDL#	3, 3, SZC	; IS CONDITION = MUX STATUS WORD ?
561	412	JMP	TCMSW	; YES
562	126000	ADC	1, 1	
563	175222	MOVZR	3, 3, SZC	; ARE STATUS BITS TO BE MASKED OUT ?
564	24072	LDA	1, MSTAT	; YES
565	173000	ADD	3, 2	
566	35000	LDA	3, 0, 2	; COMPARE SPECIFIED WORD WITH
567	167400	AND	3, 1	; ITEM IN CONDITION LIST
570	6011	TEST		
571	3776	FMCWD		; MUX CONTROL WORD WRONG
572	755	JMP	TCOND	
573	64425	TCMSW: DIA	1, MUX	; READ MUX STATUS WORD
574	175202	MOVR	3, 3, SZC	; 8 PULSE DESIRED ?
575	60125	NIDS	MUX	; YES
576	175202	MOVR	3, 3, SZC	; PORT ADDRESS TO BE INSERTED ?
577	417	JMP	TCMS1	; NO
600	34133	LDA	3, BLKL	; IF BLOCK LENGTH = 20,
601	177300	ADDS	3, 3	; LEFT-SHIFT PORT ADDRESS ONCE.
602	175101	MOVL	3, 3, SKP	; IF BLOCK LENGTH = 10,
603	151100	MOVL	2, 2	; LEFT-SHIFT PORT ADDRESS TWICE.
604	175103	MOVL	3, 3, SNC	
605	776	JMP	. -2	
606	34062	LDA	3, C7740	; MASK 7 BITS OF PORT ADDRESS
607	173400	AND	3, 2	
610	174000	COM	3, 3	
611	163400	AND	3, 0	; AND INSERT IN REF. MUX STATUS WORD
612	143000	ADD	2, 0	
613	6011	TCOMP: TEST		; IS ACTUAL MUX STATUS WORD CORRECT ?
614	4004	FMSW		; MUX STATUS WORD WRONG
615	732	JMP	TCOND	
616	34067	TCMS1: LDA	3, C140K	; TEST ONLY BITS 0 AND 1
617	101015	SNZ	0, 0	; IS LIST ENTRY = 0 ?
620	34070	LDA	3, C170K	; YES, THEN TEST BITS 0 - 3
621	167400	AND	3, 1	
622	771	JMP	TCOMP	

; ENABLE TYPE-OUT

```
623 62677 ENBTP: IORST
624 102400 SUB 0,0
625 40106 STA 0,CMD
626 24103 LDA 1,MRATE
627 125015 SNZ 1,1 ; IS A MASTER TERMINAL PCON GIVEN ?
630 1400 JMP 0,3 ; NO
631 30132 LDA 2,BASE ; SET UP MASTER TERMINAL BAUD RATE
632 71025 DOA 2,MUX
633 45001 STA 1,1,2
634 63025 DOC 0,MUX
635 25001 LDA 1,1,2
636 125112 SSP 1,1
637 403 JMP .+3 ; WAIT TILL MUX ACCEPTS PCON
640 101404 INC 0,0,SZR ; OR TIMES OUT
641 774 JMP .-4
642 60225 NI0C MUX
643 1400 C. RTN: JMP 0,3
```

; CORE CHECK - CALCULATES AND COMPARES CHECKSUMS

```
644 54115 CORC2: STA 3,PNTR ; PNTR = POINTER INTO CORE CHECK TABLE
645 4205 CRCBL: JSR FBAK
646 36115 LDA 3,@PNTR ; A3 = POINTER INTO CORE
647 20130 LDA 0,CCFLG
650 175014 SKZ 3,3 ; END OF TABLE ...
651 101014 SKZ 0,0 ; ... OR CORE CHECK DISABLED ?
652 2114 JMP @RTNAD ; YES
653 30115 LDA 2,PNTR
654 31001 LDA 2,1,2
655 126440 SUBO 1,1 ; A1 WILL ACCUMULATE CHECKSUM
656 21400 CRCWD: LDA 0,0,3
657 107100 ADDL 0,1
660 175400 INC 3,3
661 172032 SGE 3,2 ; END OF A CHECKSUM BLOCK ?
662 774 JMP CRCWD ; NO - GET NEXT WORD
663 21400 LDA 0,0,3 ; YES - TEST CHECKSUM
664 6011 TEST ; CHECKSUM ERROR ?
665 3512 PANIC ; "PANIC!! DIAGNOSTIC BEING CLOBBED"
666 4205 JSR FBAK
667 10115 ISZ PNTR
670 10115 ISZ PNTR
671 754 JMP CRCBL
```

; *** PART 1 OF DIAGNOSTIC PROGRAM ***

672 60202 NIDCO:NIOC 2
673 60225 NIDCM:NIOC MUX
674 60277 NIOC7:NIOC 77

675 62677 START: IORST ; PART 1 STARTS HERE
676 30074 LDA 2, TSO ; CLEAR TEMP. STORE AREA
677 34075 LDA 3, TSN
700 102400 SUB 0, 0
701 41000 STA 0, 0, 2
702 151400 INC 2, 2
703 156432 SGR 2, 3
704 775 JMP . -3
705 4147 JSR CORCX ; CHECK PGM CHECKSUM EXC. "LIST"
706 102620 SUBZR 0, 0 ; INITIALIZE PCWS IN "LIST"
707 2401 GO ; SO THEY WILL PASS FULL CORCK
710 2424 INLST

711 4146 STRT2: JSR CORCK
712 4205 JSR FBAK
713 4315 JSR MSG
714 4124 MTITL ; TYPE TITLE
715 4146 JSR CORCK
716 4205 JSR FBAK
717 63425 SKPBN MUX ; CHECK INITIAL STATE OF BUSY AND DONE
720 403 JMP . +3
721 6012 JSR @. FALT
722 3575 FBUSY ; FAULT: BUSY FLAG IS ON
723 63625 SKPDN MUX
724 403 JMP . +3
725 6012 JSR @. FALT
726 3601 FDONE ; DONE FLAG: -12 V. MISSING ?
727 34101 LDA 3, TOPWD ; SIZE CORE
730 24060 LDA 1, C4000
731 45400 STA 1, 0, 3
732 21400 LDA 0, 0, 3
733 54131 STA 3, CORSZ
734 136400 SUB 1, 3
735 106414 SEQ 0, 1
736 773 JMP . -5
737 4315 JSR MSG
740 4144 MCORE ; TYPE "TOP WORD USED = "
741 20131 LDA 0, CORSZ
742 4331 JSR TPOCT

; TEST MUX RESPONSE TO ALL DEVICE CODES

```
743 4205 JSR FBAK
744 20726 LDA 0,NIDCO
745 40104 STA 0,XNIOC
746 20675 LDA 0,C.RTN
747 40105 STA 0,XNIOC+1
750 10104 DEVCO: ISZ XNIOC
751 4205 JSR FBAK
752 4235 JSR OUTIM ; TURN ON REAL-TIME CLOCK
753 4104 JSR XNIOC ; GIVE NIOC TO EACH DEVICE CODE
754 24104 LDA 1,XNIOC
755 20716 LDA 0,NIDCM
756 63625 SKPDN MUX ; IS DONE FLAG STILL ON ?
757 122404 SUB 1,0,SZR ; NO: GOOD IF DEVICE CODE = 25
760 122414 SEQ 1,0 ; YES: GOOD IF DEVICE CODE <> 25
761 403 JMP .+3
762 6012 JSR @.FALT
763 3614 FDEVC

764 4205 JSR FBAK
765 24104 LDA 1,XNIOC
766 20706 LDA 0,NIDC7
767 122032 SQE 1,0 ; HAS LAST DEVICE CODE BEEN TESTED ?
770 760 JMP DEVCO ; NO
```

; TEST BUSY FLAG TURN-ON AND TURN-OFF

```
771 4205 JSR FBAK ; PUTS OUT IORST
772 63425 SKPBN MUX
773 403 JMP .+3 ; BUSY SHOULD BE OFF
774 6012 XBUSY: JSR @.FALT
775 3575 FBUSY

776 63125 DOCS 0,MUX ; START MUX
777 63425 SKPBN MUX ; BUSY SHOULD BE ON
1000 774 JMP XBUSY
1001 62125 DOBS 0,MUX ; MUX PAUSE
1002 63525 SKPBZ MUX
1003 771 JMP XBUSY
1004 63225 DOCC 0,MUX ; TURN MUX ON AND RIGHT BACK OFF AGAIN
1005 63525 SKPBZ MUX
1006 766 JMP XBUSY
1007 63025 DOC 0,MUX ; MUX CONTINUE
1010 63425 SKPBN MUX
1011 763 JMP XBUSY
1012 62677 IORST
1013 63525 SKPBZ MUX
1014 760 JMP XBUSY
1015 4146 JSR CORCK ; MAKE SURE MUX HASN'T CLOBBERED PGM
```

; TEST REAL-TIME CLOCK BY USING TELETYPE AS A REFERENCE

```

1016 4205 JSR FBAK
1017 4235 JSR OUTIM
1020 152400 SUB 2,2
1021 71111 DOAS 2,TTO
1022 4226 JSR EIMUX ; ENABLE INTPTS FROM MUX
1023 63511 SKPBZ TTO ; EACH REAL-TIME INTPT INCREMENTS A2
1024 777 JMP .-1
1025 60277 INTDS
1026 145004 MOV 2,1,SZR
1027 403 JMP .+3
1030 6012 JSR @.FALT
1031 3635 FNINT ; NO INTERRUPTS OCCURRED ?!
1032 20102 LDA 0,N.RTC ; THERE SHD HAVE BEEN 10 OR 11 INTPTS
1033 106654 SUBOR# 0,1,SZR
1034 125235 MOVZR# 1,1,SNR ; OR JUST 1
1035 403 JMP .+3
1036 6012 JSR @.FALT
1037 3672 FRTC ; REAL-TIME CLOCK WRONG

```

; DETERMINE CPU SPEED BY COUNTING THE NUMBER OF "STANDARD TIMING LOOPS"
; IT CAN MAKE IN 10 MILLISECONDS

```

1040 4205 JSR FBAK
1041 4315 JSR MSG
1042 4155 MSPED ; "EFFECTIVE MEMORY CYCLE TIME = "
1043 4235 JSR OUTIM
1044 4226 JSR EIMUX ; ENABLE INTERRUPT
1045 176440 SUBO 3,3 ; CLEAR A3 AND CARRY (= 17-BIT ZERO)
1046 152440 SUBO 2,2
1047 25000 LDA 1,0,2 ; \
1050 151014 SKZ 2,2 ; \
1051 410 JMP SPEED ; \
1052 22400 LDA 0,@ ; \ STANDARD TIMING LOOP
1053 22400 LDA 0,@ ; / (11 EFFECTIVE MEMORY CYCLES)
1054 175405 INC 3,3,SNR ; /
1055 101002 MOV 0,0,SZC ; /
1056 771 JMP .-7 ; /
1057 6012 JSR @.FALT
1060 3635 FNINT

```

```

1061 175200 SPEED: MOVR 3,3 ; A3 = 1/2 # LOOPS/10 MSEC
1062 54120 STA 3,TSCPU
1063 175220 MOVZR 3,3 ; A3 = 1/4 # LOOPS/10 MSEC
1064 30431 LDA 2,NMCP ; DIVIDE A3 INTO NMCP
1065 126000 ADC 1,1
1066 125400 INC 1,1 ; A1 = NMCP/A3 = MEM. CYCLE TIME/10
1067 172422 SUBZ 3,2,SZC
1070 776 JMP .-2
1071 4352 JSR TPDEC ; TYPE IT AS A DECIMAL NUMBER
1072 4315 JSR MSG
1073 4170 MNSEC ; "0 NANoseconds"
1074 4205 JSR FBAK
1075 4421 JSR TMUL ; CALCULATE TIME COUNTS FOR THIS CPU

```

; REFERENCE TIME CONSTANTS -- L (=256) REPRESENTS 10 MSEC

	12	.RDX	10	
1076	40	L/8		; 1.25 MSEC
1077	40	L/8		
1100	100	L/4		; 2.5 MSEC
1101	100	L/4		
1102	414	L*21/20		; 10 MSEC +- 5%
1103	31	L*2/20		
1104	4000	L*8		; 50 MSEC (MUX'S SAMPLING TIME)
1105	2000	L*4		; (80 MSEC MAX, 40 MSEC MIN)
1106	6000	L*12		; .1 SECOND +- 10%
1107	2000	L*4		
1110	16000	L*28		; .2 SECOND (.28 MAX, .16 MIN)
1111	6000	L*12		
1112	26000	L*44		; .4 SECOND (.44 MAX, .4 MIN)
1113	2000	L*4		
1114	0	0		; END OF LIST
1115	54300	NMCPL: 25000/11*10		; 10^6/ # MEM. CYCLES PER TIM. LOOP/ 4
	10	.RDX	8	

; MULTIPLY THE NUMBER OF TIMING LOOPS THIS CPU MADE IN 10 MSEC
; BY EACH OF THE ABOVE CONSTANTS, IN WHICH THE BINARY POINT
; IS IN THE CENTER OF THE WORD (I.E. OCTAL 400 MEANS 1.0),
; AND STORE ITS NEGATIVE IN VARIABLE TIME COUNTS TABLE (@.TVAR)

1116	54121	TMUL: STA	3, TSTK	; POINTER TO ABOVE TIME CONSTANTS
1117	20076	LDA	0, .TVAR	
1120	40122	STA	0, TSTV	; POINTER TO VAR. TIME COUNT TABLE
1121	26121	TMUL1: LDA	1, @TSTK	
1122	125015	SNZ	1, 1	; END OF LIST ?
1123	425	JMP	FCORE	; YES
1124	30120	LDA	2, TSCPU	
1125	176520	SUBZL	3, 3	
1126	102400	SUB	0, 0	
1127	125203	MOVR	1, 1, SNC	
1130	101201	MOVR	0, 0, SKP	
1131	143220	ADDZR	2, 0	
1132	177004	ADD	3, 3, SZR	
1133	774	JMP	. -4	
1134	125260	MOVCR	1, 1	
1135	34071	LDA	3, C377L	
1136	163414	AND#	3, 0, SZR	
1137	63077	HALT		; CPU TOO FAST
1140	167400	AND	3, 1	
1141	123300	ADDS	1, 0	
1142	100000	COM	0, 0	
1143	42122	STA	0, @TSTV	; STORE THE PRODUCT
1144	10122	ISZ	TSTV	; INCREMENT THE 2 TABLE POINTERS
1145	10121	ISZ	TSTK	
1146	753	JMP	TMUL1	; MULTIPLY NEXT TABLE ENTRY
1147	77204	CKS3: 77204		; CHECKSUM

; FILL CORE WITH 40360 (PCON FOR 110 BAUD, OR AUTO INPUT IF ICW)

1150	4561	FCORE:	JSR	FILCO	
1151	30100		LDA	2, BOTWD	; RECHECK CORE FOR ALL 40360
1152	4570		JSR	SRCNX+1	
1153	404		JMP	. +4	; ALL 40360 UP TO TOP OF CORE
1154	63077		HALT		; NEVER USED
1155	6012		JSR	@. FALT	
1156	3526		FGARB		; "GARBAGE COMING IN"

; START MUX AND TEST FOR NO UNAUTHORIZED INTERRUPTS

1157	4205		JSR	FBAK	
1160	102000		ADC	0, 0	; MASK OUT ALL DEVICES
1161	62077		MSKO	0	
1162	4403		JSR	. +3	; SET UP INTPT FOR FAULT
1163	6012	INTF:	JSR	@. FALT	
1164	3624		FINT		; ILLEGAL INTERRUPT OCCURRED
1165	54001		STA	3, 1	; STORE INTERRUPT SERVICE ADDRESS
1166	63325		DOCP	0, MUX	; START MUX
1167	60177		INTEN		; TEST FOR NO INTERRUPT
1170	4364		JSR	TIMCK	
1171	32		T. 2SE		
1172	177777		-1		
1173	60277		INTDS		

; DETERMINE MUX CONTROL AREA BASE ADDRESS

1174	4146		JSR	CORCK	
1175	4205		JSR	FBAK	
1176	30100		LDA	2, BOTWD	
1177	4543		JSR	SRCNX+1	; TYPES "GARBAGE" IF NOT OCW = 14X360
1200	6012		JSR	@. FALT	; TOP OF CORE -- NO MUX ACTION
1201	3534		FNOMX		
1202	50132		STA	2, BASE	; SRCNX NORMALLY RETURNS HERE
1203	50136		STA	2, XMTR	
1204	4315		JSR	MSG	
1205	4175		MBASE		; "MUX CONTROL AREA BASE = "
1206	20132		LDA	0, BASE	
1207	4331		JSR	TPOCT	

; DETERMINE MUX CONTROL BLOCK LENGTH

1210	30132	LDA	2, BASE	
1211	4530	JSR	SRCNX	
1212	6012	JSR	@, FALT	
1213	3541	FBLKL		; ONLY ONE PORT ?!
1214	50137	STA	2, RCVR	
1215	50134	STA	2, LPORT	
1216	20132	LDA	0, BASE	
1217	112400	SUB	0, 2	
1220	145000	MOV	2, 1	
1221	34053	LDA	3, C40	
1222	171220	MOVZR	3, 2	; BLOCK LENGTH MUST = 10, 20, OR 40
1223	141220	MOVZR	2, 0	
1224	136414	SEQ	1, 3	
1225	132415	SNE	1, 2	
1226	403	JMP	. +3	
1227	6011	TEST		
1230	3541	FBLKL		; "CANNOT DET. VALID BLOCK LENGTH"
1231	44133	STA	1, BLKL	
1232	4315	JSR	MSG	
1233	4206	MBLKL		; "BLOCK LENGTH = "
1234	20133	LDA	0, BLKL	
1235	4331	JSR	TPOCT	

; DETERMINE NUMBER OF PORTS

1236	20156	LDA	0, C2	
1237	40120	STA	0, TS	; INITIALIZE PORT COUNT
1240	30134	NPTLP: LDA	2, LPORT	
1241	4500	JSR	SRCNX	; SEARCH FOR NEXT PORT
1242	413	JMP	NO. PT	; TOP OF CORE: PORT COUNT COMPLETED
1243	63077	HALT		
1244	10120	ISZ	TS	; INCREMENT PORT COUNT
1245	24134	LDA	1, LPORT	; CONFIRM SPACING BETWEEN PORTS
1246	50134	STA	2, LPORT	
1247	132400	SUB	1, 2	
1250	145000	MOV	2, 1	
1251	20133	LDA	0, BLKL	
1252	6011	TEST		
1253	3555	FNSEQ		; "PORTS NOT IN SEQUENCE"
1254	764	JMP	NPTLP	
1255	4315	NO. PT: JSR	MSG	
1256	4214	MNPTS		; "NUMBER OF PORTS = "
1257	20120	LDA	0, TS	
1260	4331	JSR	TPOCT	

; TEST DOA CAPABILITY - I. E. CHANGE PORT CONTROL AREA BY SOFTWARE

```
1261 4205 JSR FBAK
1262 102120 ADCZL 0,0
1263 40124 STA 0,SWICH
1264 20444 LDA 0,C65K
1265 24131 LDA 1,CORSZ ;CALC. HIGHEST POSSIBLE ADDRESS
1266 152220 ADCZR 2,2
1267 143400 AND 2,0
1270 151220 MOVZR 2,2
1271 106033 SLS 0,1
1272 775 JMP -3
1273 40121 STDOA: STA 0,TS+1
1274 61025 DOA 0,MUX ;OUTPUT THE DESIRED PCA ADDRESS
1275 4434 JSR FILCO ;FILL CORE WITH 40360
1276 63025 DOC 0,MUX ;START MUX
1277 61025 DOA 0,MUX ;SHOULD HAVE NO EFFECT BEC. MUX IS ON
1300 4364 JSR TIMCK
1301 20 T1MS
1302 177777 -1
1303 30100 LDA 2,BOTWD
1304 4436 JSR SRCNX+1 ;SEARCH CORE FOR FIRST CHANGED WORD
1305 6012 JSR @.FALT ;TOP OF CORE - NO MUX ACTION !?
1306 3771 FDOA
1307 20121 LDA 0,TS+1
1310 145000 MOV 2,1
1311 6011 TEST ;DID WE FIND THE RIGHT PLACE ?
1312 3771 FDOA ; NO, DOA CAPABILITY NOT WORKING
1313 20414 LDA 0,C12K4 ;REPEAT WITH A DIFFERENT ADDRESS
1314 10124 ISZ SWICH
1315 756 JMP STDOA
1316 4205 JSR FBAK
1317 4315 JSR MSG
1320 4223 MECMD ;"ENTER COMMAND (PRESS H FOR HELP)"
1321 20106 LDA 0,CMD
1322 103112 ADDL# 0,0,SZC ;IS CMD = F ?
1323 2542 JMP @.PT2 ; YES, START PART 2
1324 456 JMP WAIT

1325 4205 JSR FBAK ;RETURN TO WAIT IF C IS PRESSED
1326 454 JMP WAIT

1327 12400 C12K4: 12400
1330 65000 C65K: 65000
```

```
1331 30100 FILCO: LDA 2, BOTWD ; FILL CORE WITH 40360
1332 24131 LDA 1, CORSZ
1333 20065 LDA 0, C4OKX
1334 151400 INC 2, 2
1335 41000 STA 0, 0, 2
1336 146032 SGE 2, 1
1337 775 JMP . -3
1340 1400 JMP 0, 3
```

```
; SEARCH CORE (STARTING AT A2+2) FOR NEXT WORD NOT EQUAL TO 40360.
; NON-SKIP RETURN IF TOP OF CORE (CORSZ) IS REACHED; DOUBLE-SKIP RETURN
; (WITH A2 = .OCW - 1) IF AN OCW IS FOUND CONTAINING 140360 OR 146760.
; IN ALL OTHER CASES IT TYPES "GARBAGE COMING IN".
```

```
1341 151400 SRCNX: INC 2, 2 ; ENTRY HERE STARTS SEARCH AT A2+2
1342 54114 STA 3, RTNAD ; ENTRY HERE STARTS SEARCH AT A2+1
1343 50115 STA 2, PNTR
1344 4205 JSR FBAK
1345 10115 ISZ PNTR
1346 30115 LDA 2, PNTR
1347 34131 LDA 3, CORSZ
1350 20065 LDA 0, C4OKX
1351 156433 SLE 2, 3
1352 2114 JMP @RTNAD ; TOP OF CORE
1353 25000 LDA 1, 0, 2
1354 106415 SNE 0, 1 ; FOUND A CHANGED WORD ?
1355 766 JMP SRCNX+2 ; NO
1356 34044 LDA 3, C7
1357 157404 AND 2, 3, SZR ; IS IT AN ICW . . .
1360 175224 MOVZR 3, 3, SZR ; OR NOT AN OCW ?
1361 406 JMP S, GRB ; YES - TYPE "GARBAGE"
1362 34072 LDA 3, MSTAT ; YES - MASK IT
1363 137400 AND 1, 3
1364 20413 LDA 0, GOCW
1365 162415 SNE 3, 0 ; GOOD VALUE ?
1366 403 JMP . +3 ; YES
1367 6012 S, GRB: JSR @, FALT ; NO - TYPE:
1370 3526 FGARB ; "GARBAGE COMING IN "
1371 34114 LDA 3, RTNAD ; YES - DOUBLE-SKIP RETURN
1372 150400 NEG 2, 2 ; DECREMENT A2
1373 150000 COM 2, 2
1374 1402 JMP 2, 3

1375 4205 JSR FBAK ; RE-ENTRY IF "C" AFTER "GARBAGE"
1376 746 JMP SRCNX+3
```

```
1377 140360 GOCW: 140360 ; EXPECTED VALUE OF OCW
```

; WAIT FOR TELETYPE INPUT

1400	4324	ABORT:	JSR	TYPE	
1401	56000		"*L		
1402	6013	WAIT:	JSR	@. ENTP	; ENABLE TYPEDOUT
1403	102520		SUBZL	0, 0	
1404	40107		STA	0, SPFLQ	
1405	4324		JSR	TYPE	; CR, LF, >
1406	6412		15*L+12		
1407	37000		">*L		
1410	102400	WAIT1:	SUB	0, 0	
1411	40120		STA	0, TS	; INITIALIZE NUMERIC ENTRY FLAG
1412	101125	WAITN:	MOVZL	0, 0, SNR	; FLASH CARRY LIGHT
1413	20545		LDA	0, PATRN	; IN A RECOGNIZABLE PATTERN
1414	24034		LDA	1, T. 48E	
1415	36400		LDA	3, @.	
1416	36400		LDA	3, @.	
1417	125405		INC	1, 1, SNR	
1420	772		JMP	. -6	
1421	63610		SKPDN	TTI	
1422	773		JMP	. -5	
1423	60610		DIAC	0, TTI	
1424	4522		JSR	BRESC	; RETURN TO WAIT ON ESCAPE
1425	61111		DOAS	0, TTO	; ECHO IT
1426	63511		SKPBZ	TTO	; WAIT TILL TTY DONE BEFORE PROCEEDING
1427	777		JMP	. -1	
1430	60211		NIDC	TTO	
1431	176400		SUB	3, 3	; CLEAR COMMAND WORD
1432	54106		STA	3, CMD	
1433	4475		JSR	BRNCH	
1434	102		"B		; COMMAND BRANCH TABLE
1435	675		START		
1436	103		"C		
1437	1712		IN. C		
1440	104		"D		
1441	1642		IN. D		
1442	105		"E		
1443	1742		IN. E		
1444	106		"F		
1445	1710		IN. F		
1446	110		"H		
1447	2034		IN. H		
1450	111		"I		
1451	367		IPL		
1452	115		"M		
1453	1722		IN. M		
1454	116		"N		
1455	3206		IN. N		
1456	117		"O		
1457	4322		. INO-1		
1460	121		"Q		
1461	4326		IN. Q		
1462	122		"R		
1463	1632		IN. R		
1464	123		"S		
1465	2057	. PT2:	PART2		
1466	124		"T		
1467	2056		IN. T		

1470	125	"U		
1471	1763	IN. U		
1472	130	"X		
1473	1772	IN. X		
1474	131	"Y		
1475	4324	. INY-1		
1476	132	"Z		
1477	2007	IN. Z		
1500	54	",		
1501	1561	INCOM		
1502	15	15		; CARRIAGE RETURN
1503	1564	IN. CR		
1504	50	"{		
1505	1620	INCMT		
1506	40	40		; SPACE
1507	1701	IN. SP		
1510	20	20		; CONTROL P
1511	2037	IN. CP		
1512	0	0		
1513	30054	LDA	2, C60	
1514	34055	LDA	3, C67	
1515	112033	SLS	0, 2	; IS AO AN OCTAL DIGIT ?
1516	116433	SLE	0, 3	
1517	661	JMP	ABORT	; NO
1520	142540	SUBOL	2, 0	; YES -- SHIFT IT INTO TS
1521	24120	LDA	1, TS	
1522	125120	MOVZL	1, 1	
1523	125120	MOVZL	1, 1	
1524	127120	ADDZL	1, 1	
1525	107240	ADDOR	0, 1	; MSB OF TS = NO. AVAILABLE FLAG
1526	44120	STA	1, TS	
1527	663	JMP	WAITN	
1530	24057	BRNCH: LDA	1, C177	
1531	123400	AND	1, 0	
1532	125620	INCZR	1, 1	
1533	107400	AND	0, 1	
1534	124240	COMOR	1, 1	; IS INPUT A LETTER ?
1535	123400	AND	1, 0	; YES, CHANGE LOWER CASE TO CAPITAL
1536	25400	BRNC2: LDA	1, 0, 3	
1537	125015	SNZ	1, 1	
1540	1401	JMP	1, 3	
1541	106415	SNE	0, 1	
1542	3401	JMP	@1, 3	
1543	175400	INC	3, 3	
1544	175400	INC	3, 3	
1545	771	JMP	BRNC2	
1546	54121	BRESC: STA	3, TS+1	; BRANCH ON ESCAPE
1547	4761	JSR	BRNCH	
1550	33	33		; ESCAPE
1551	1402	WAIT		
1552	4	4		; CTRL D
1553	1402	WAIT		
1554	135	135		; ALT MODE
1555	1402	WAIT		
1556	0	0		
1557	2121	JMP	@TS+1	

```
1560 10425 PATRN: 10425 ; CARRY LIGHT FLASH PATTERN

1561 4416 INCOM: JSR CNVRT ; INPUT = COMMA
1562 50136 STA 2, XMTR
1563 625 JMP WAIT1

1564 4413 IN. CR: JSR CNVRT ; INPUT = CARRIAGE RETURN
1565 50137 STA 2, RCVR
1566 4315 JSR MSG ; TYPE OUT XMTG & RCVG PORTS
1567 4237 MXMTR
1570 20136 LDA 0, XMTR
1571 4331 JSR TPOCT
1572 4315 JSR MSG
1573 4243 MRCVR
1574 20137 LDA 0, RCVR
1575 4331 JSR TPOCT
1576 604 JMP WAIT

1577 30120 CNVRT: LDA 2, TS ; CONVERT # IN TS TO A PORT ADDRESS
1600 151103 MOVL 2, 2, SNC ; MSB = # AVAIL. FLAG
1601 1401 JMP 1, 3 ; SKIP RETURN IF NO # IN TS
1602 151220 MOVZR 2, 2
1603 24100 LDA 1, BOTWD ; IF # > BOTWD, IT'S A PORT ADDR.
1604 146433 SLE 2, 1
1605 1400 JMP 0, 3
1606 24057 LDA 1, C177 ; PORT NUMBER MUST BE < 128
1607 146433 SLE 2, 1
1610 2010 JMP @. ABOR
1611 140000 COM 2, 0
1612 30132 LDA 2, BASE ; MULT. PORT # BY BLKL & ADD TO BASE
1613 24133 LDA 1, BLKL
1614 101405 INC 0, 0, SNR
1615 1400 JMP 0, 3
1616 133000 ADD 1, 2
1617 775 JMP . -3

1620 63610 INCMT: SKPDN TTI ; COMMENT MODE
1621 777 JMP . -1
1622 60510 DIAS 0, TTI
1623 4723 JSR BRESC ; GO TO WAIT IF IT WAS ESCAPE
1624 61111 DOAS 0, TTO ; ECHO EVERY CHARACTER
1625 4703 JSR BRNCH ; GO TO WAIT IF ")"
1626 51 ")
1627 1403 WAIT+1
1630 0 0
1631 767 JMP INCMT ; OTHERWISE STAY IN COMMENT MODE

1632 102620 IN. R: SUBZR 0, 0 ; MSB OF CMD = REPEAT
1633 101400 INC 0, 0 ; SET "INHIBIT TYPE-OUT" FLAG
1634 40106 STA 0, CMD
1635 20120 LDA 0, TS
1636 126620 SUBZR 1, 1
1637 123022 ADDZ 1, 0, SZC ; WAS A NUMBER TYPED IN ?
1640 40117 STA 0, FBAK ; YES, REPEAT FROM THERE
1641 4205 JSRFB: JSR FBAK
```

; INPUT = D: DUMP 4 LINES OF DATA. FIRST, ACCUMULATORS AND CERTAIN
 ; KEY VARIABLES (RTNAD,PNTR,COUNT,.FBAK). SECOND, MUX CONTROL WORDS
 ; OF TRANSMITTING PORT. THIRD, MUX CONTROL WORDS OF RECEIVING PORT.
 ; FOURTH, LAST AUTOMATIC BUFFER USED (ADDRESS AND CONTENTS).

```

1642 20073 IN. D: LDA 0, AO
1643 4412 JSR DUMP
1644 20136 LDA 0, XMTR
1645 4410 JSR DUMP
1646 20137 LDA 0, RCVR
1647 24136 LDA 1, XMTR
1650 106414 SEQ 0, 1
1651 4404 JSR DUMP
1652 20141 LDA 0, BFR
1653 4402 JSR DUMP
1654 2006 JMP @. WAIT

1655 54122 DUMP: STA 3, TS+2
1656 40121 STA 0, TS+1
1657 10107 ISZ SPFLG ; SET SPACE FLAG TO 2
1660 4324 JSR TYPE ; CR, LF
1661 175673 @"_H*F+"^H*F+"^-H
1662 20121 LDA 0, TS+1
1663 4332 JSR TPOCL
1664 4324 JSR TYPE
1665 35000 " : *L
1666 20045 LDA 0, C10
1667 40120 STA 0, TS
1670 4324 DNEXT: JSR TYPE
1671 20000 " *L
1672 22121 LDA 0, @TS+1
1673 10121 ISZ TS+1
1674 4332 JSR TPOCL
1675 14120 DSZ TS
1676 772 JMP DNEXT
1677 6013 CDUMP: JSR @. ENTP
1700 2122 JMP @TS+2

1701 10106 IN. SP: ISZ CMD ; SET "INHIBIT TYPE-OUT" FLAG
1702 20107 LDA 0, SPFLG
1703 101015 SNZ 0, 0 ; IS SPACE FLAG SET ?
1704 2117 JMP @. FBAK ; NO, RETURN TO FALLBACK POINT
1705 101235 MOVZR# 0, 0, SNR ; YES - WERE WE IN DUMP ?
1706 2006 JMP @. WAIT ; NO - ENABLE TYPE & GO TO "WAIT"
1707 770 JMP CDUMP ; YES

1710 20411 IN. F: LDA 0, CMDF ; RUN TEST "FOREVER"
1711 40106 STA 0, CMD
1712 34117 IN. C: LDA 3, . FBAK ; CONTINUE FROM NEXT FALLBACK POINT
1713 24726 LDA 1, JSRFB
1714 175400 INC 3, 3
1715 21400 LDA 0, 0, 3
1716 106414 SEQ 0, 1
1717 775 JMP . -3
1720 1400 JMP 0, 3

1721 40001 CMDF: 40001 ; "FOREVER" COMMAND

```

; INPUT = M: SENDS MAIL BETWEEN TTY AND SELECTED MUX PORT
; TTY INPUT ==> XMTR
; XMTR INPUT ==> TTY
; IF A NUMBER WAS TYPED BEFORE M, IT IS SENT TO PORT AS AN OCW (FOR PCON)

```
1722 4503 IN. M: JSR     SETUP
1723 126400      SUB     1, 1
1724 62025      DOB     0, MUX      ; PAUSE MUX
1725 21000      LDA     0, 0, 2    ; PICK UP AND CLEAR ICW
1726 45000      STA     1, 0, 2
1727 63025      DOC     0, MUX      ; CONTINUE MUX
1730 101112     SSP     0, 0        ; WAS ANY MUX INPUT RECEIVED ?
1731 61111      DOAS    0, TTD      ; YES - TYPE IT
1732 63610      SKPDN   TTI        ; WAS ANY TTY INPUT RECEIVED ?
1733 770        JMP     IN. M+1      ; NO
1734 60610      DIAC    0, TTI      ; YES - READ IT IN
1735 4611       JSR     BRESK      ; EXIT IF IT WAS ESCAPE
1736 24066      LDA     1, C6OK     ; OTHERWISE SEND IT TO MUX
1737 123000     ADD     1, 0
1740 41001      STA     0, OCW. , 2
1741 762        JMP     IN. M+1
```

; INPUT = E: AUTO ECHO. SETS UP 8-BYTE AUTO INPUT BUFFER WITH
; ICW = AUTO ECHO. WHEN "DONE", STARTS AUTO OUTPUT FROM
; SAME BUFFER WITH SPECIAL CHARACTER INTERRUPT REQUEST

```
1742 4221 IN. E: JSR     EITTY
1743 4462      JSR     SETUP
1744 20140     LDA     0, IBP
1745 41004     STA     0, IBP. , 2
1746 41005     STA     0, OBP. , 2
1747 24045     LDA     1, C10
1750 123000     ADD     1, 0
1751 41006     STA     0, LIB. , 2
1752 41007     STA     0, LOB. , 2
1753 20066     LDA     0, C6OK     ; ICW FOR AUTO INPUT WITH ECHO
1754 41000     STA     0, ICW. , 2
1755 21000     LDA     0, ICW. , 2 ; WAIT FOR BUFFER FULL (OR PAR. ERR. )
1756 101113     SSN     0, 0
1757 776       JMP     -2
1760 4253      JSR     STOCW
1761 30301     30301      ; OCW FOR AUTO OUTPUT W SPL INTPT REQ
1762 762       JMP     IN. E+2
```

; INPUT = U: JUMP TO DEBUG, IF IT'S THERE

```
1763 22405 IN. U: LDA     @. DEBUG
1764 24405     LDA     1, CDEBUG
1765 106414    SEQ     0, 1        ; IS THE WORD AT .DEBUG = 524 ?
1766 2010      JMP     @. ABOR      ; NO
1767 2401      JMP     @. DEBUG     ; YES

1770 6000 .DEBUG: 6000
1771 524  CDEBUG: 524
```


; INPUT = X: READ SWITCHES AND ENTER INTO TRANSMITTING PORT'S DCW;
; LOAD OTHER CONTROL WORDS FROM TABLE

1772	4221	IN. X:	JSR	EITTY	; ENABLE INTERRUPTS FROM TELETYPE
1773	4432		JSR	SETUP	
1774	34137		LDA	3, RCVR	
1775	45400		STA	1, ICW, 3	
1776	20142		LDA	0, XPCON	
1777	41401		STA	0, DCW, 3	
2000	20145		LDA	0, XLOB	
2001	41007		STA	0, LOB, 2	
2002	20144		LDA	0, XOBP	
2003	41005		STA	0, OBP, 2	
2004	60477		READS	0	; READ DCW FROM SWITCHES
2005	41001		STA	0, DCW, 2	
2006	774		JMP	. -4	

; INPUT = Z: A SIMPLE TEST TO EXERCISE ALL CPU I/O CONTROL LINES.
; USEFUL FOR DEBUGGING WHEN "BIG LOOP" IS DEAD. FOR VARIABLE
; REP RATE SET SWITCH 0 UP.

2007	61477	IN. Z:	DIB	0, CPU	; = INTA
2010	64525		DIAS	1, MUX	
2011	62677		IORST		
2012	4221		JSR	EITTY	; SENDS OUT MSKO PULSE
2013	60477		DIA	0, CPU	; = READS 0
2014	72452		DIC	2, 77-MUX	
2015	30132		LDA	2, BASE	
2016	71025		DOA	2, MUX	
2017	62225		DOBC	0, MUX	
2020	63325		DOCP	0, MUX	; STARTS MUX AND RTC
2021	101112		SSP	0, 0	; IS SWITCH 0 UP ?
2022	101405		INC	0, 0, SNR	; YES, COUNT TO 0 BEFORE RETURNING
2023	764		JMP	IN. Z	; NO, RETURN IMMEDIATELY
2024	776		JMP	. -2	

2025	30136	SETUP:	LDA	2, XMTR	
2026	20120		LDA	0, TS	
2027	126620		SUBZR	1, 1	
2030	123022		ADDZ	1, 0, SZC	; WAS A NUMBER TYPED ?
2031	41001		STA	0, DCW, 2	; YES - SEND IT TO XMTR
2032	63025		DOC	0, MUX	
2033	1400		JMP	0, 3	

2034	4315	IN. H:	JSR	MSG	; TYPE "HELP" MESSAGE
2035	4247		MHELP		
2036	2006		JMP	@. WAIT	

2037	4324	IN. CP:	JSR	TYPE	; INPUT = CONTROL P
2040	22400		"%*L		; ECHO A % AND
2041	10130		ISZ	CCFLG	; DISABLE CORE CHECK
2042	2006		JMP	@. WAIT	

2043	160010	CKS4:	160010		; CHECKSUM
------	--------	-------	--------	--	------------

; *** PART 2 OF DIAGNOSTIC PROGRAM ***

2044 456 NTSTS: 1+6+100+60+40*2 ; 2 * NUMBER OF INDIVIDUAL SUBTESTS
2045 102400 CNVTP: SUB 0, 0 ; CONVERT PORT ADDRESS INTO PORT NO.
2046 30132 LDA 2, BASE
2047 146405 SUB 2, 1, SNR
2050 331 JMP TPOCT ; TYPE PORT NUMBER AND RETURN
2051 125112 SSP 1, 1
2052 2010 JMP @. ABOR
2053 101400 INC 0, 0
2054 30133 LDA 2, BLKL
2055 772 JMP CNVTP+2

2056 63077 IN. T: HALT ; TIME FOR CONNECTIONS IN TTY MODE

2057 62677 PART2: IORST ; PART 2 STARTS HERE

2060 4147 JSR CORCX

; COMPUTATION OF INITIAL IBP AND "DELTA", I. E. AMOUNT BY WHICH THE INPUT
; BUFFER IS ADVANCED AFTER EACH TEST, IN ORDER TO TEST AS MUCH OF
; THE ADDRESSING LOGIC AS POSSIBLE

2061 4205 JSR FBAK
2062 20100 LDA 0, BOTWD
2063 24132 LDA 1, BASE
2064 106400 SUB 0, 1 ; SPACE BET. PGM & FIRST PORT
2065 30134 LDA 2, LPORT
2066 34133 LDA 3, BLKL
2067 173000 ADD 3, 2
2070 34131 LDA 3, CORSZ
2071 156423 SUBZ 2, 3, SNC ; SPACE BET. LAST PORT & TOP OF CORE
2072 176400 SUB 3, 3 ; IF RESULT < 0, MAKE IT 0
2073 136432 SGR 1, 3 ; CHOOSE THE BIGGER SPACE
2074 141000 MOV 2, 0
2075 136432 SGR 1, 3
2076 165000 MOV 3, 1
2077 101140 MOVOL 0, 0 ; CALCULATE FIRST IBP
2100 40140 STA 0, IBP
2101 20743 LDA 0, NTSTS ; DIVIDE SPACE BY NO. OF TESTS
2102 152401 SUB 2, 2, SKP
2103 151400 INC 2, 2
2104 106422 SUBZ 0, 1, SZC
2105 776 JMP . -2
2106 150520 NEGZL 2, 2 ; MAKE DELTA AN ODD NO. OF WORDS
2107 150120 COMZL 2, 2 ; FOR MAX. ADDRESS VARIETY
2110 50135 STA 2, DELTA
2111 102400 SUB 0, 0 ; CLEAR UPPER CORE
2112 30100 LDA 2, BOTWD
2113 34131 LDA 3, CORSZ
2114 151400 INC 2, 2
2115 41000 STA 0, 0, 2
2116 156032 SGE 2, 3
2117 775 JMP . -3

```
2120 4315 JSR MSG ; TYPE XMTR AND RCVR PORT NO. S
2121 64 C2OK ; OUTPUTS A SPACE
2122 24136 LDA 1, XMTR
2123 4722 JSR CNVTP
2124 4324 JSR TYPE
2125 26000 ", *L
2126 24137 LDA 1, RCVR
2127 4716 JSR CNVTP
```

; CHECK THAT XMTR DEVICE CONTROL AND STATUS ARE CONNECTED

```
2130 4205 JSR FBAK ; IORST SETS DEVICE CONTROL HIGH
2131 41001 STA 0, OCW, 2 ; ZERO OUT DEVICE STATUS IN OCW
2132 63025 DDC 0, MUX ; TURN ON MUX
2133 4364 JSR TIMCK ; WAIT ONE MUX CYCLE TIME (50 MS)
2134 26 T50MS
2135 177777 -1

2136 25001 LDA 1, OCW, 2
2137 125014 SKZ 1, 1 ; IS OCW STILL 0 ?
2140 403 JMP . +3 ; NO, ASSUME OK
2141 6012 JSR @. FALT
2142 4012 FSTAT ; "DEVICE STATUS (CONN. XMTR 8 TO 20)"
```

; TEST DEVICE CONTROL AND DEVICE STATUS LOGIC

```
2143 4205 JSR FBAK ; GIVES IORST PULSE TO MUX
2144 41001 STA 0, OCW, 2 ; CLEAR OCW OF XMTR
2145 4235 JSR OUTIM ; WAIT TILL JUST AFTER R-T INTPT.
2146 63025 DDC 0, MUX ; START MUX
2147 4353 JSR OUTCK ; WAIT FOR OUTDONE BIT TO BE POSTED
2150 20 T1MS
2151 4 OCW. T
2152 104400 104400 ; STATUS CHANGE: NEW STATUS = 1
2153 100000 MSW. T
2154 110000 110000
2155 177777 -1

2156 4364 JSR TIMCK ; WAIT 10 MILLI-SECONDS
2157 24 T10MS
2160 4 OCW. T
2161 104400 104400 ; 2D STATUS BIT HAS NOT YET CHANGED
2162 177777 -1

2163 4364 JSR TIMCK ; WAIT ONE MUX CYCLE (50 MS NOM.)
2164 26 T50MS
2165 4 OCW. T
2166 106400 106400 ; NOW BOTH STATUS BITS = 1
2167 100002 MSW. T!S
2170 10000 010000
2171 177777 -1
```

```
2172 4261 JSR STOCU ; CLEAR OUT STATUS CHG & DONE BITS
2173 6000 6000
2174 4364 JSR TIMCK ; WAIT ANOTHER 50 MSEC
2175 26 T50MS
2176 4 OCW. T
2177 6000 6000 ; NO CHANGE IN OCW
2200 100000 MSW. T
2201 100000 100000
2202 177777 -1

2203 4261 JSR STOCU ; SET DEVICE CONTROL BIT TO 0
2204 46360 46360
2205 4353 JSR OUTCK ; WAIT FOR OUT DONE BIT
2206 20 T1MS
2207 4 OCW. T
2210 146360 146360 ; STATUS HAS NOT YET CHANGED
2211 100000 MSW. T
2212 10000 010000
2213 177777 -1

2214 20061 LDA 0, C6000 ; CLEAR OUTPUT DONE BIT
2215 41001 STA 0, OCW. , 2
2216 4353 JSR OUTCK ; WAIT FOR NEXT DONE BIT
2217 26 T50MS
2220 4 OCW. T
2221 102400 102400 ; NEW STATUS CHANGED TO 0
2222 100000 MSW. T
2223 110000 110000
2224 177777 -1

2225 4364 JSR TIMCK ; WAIT ANOTHER MUX CYCLE
2226 26 T50MS
2227 4 OCW. T
2230 100400 100400 ; NOW BOTH STATUS BITS ARE 0
2231 177777 -1

2232 4253 JSR STOCW ; NOW TURN DEVICE CONTROL BACK ON
2233 50360 50360
2234 4353 JSR OUTCK ; WAIT TILL PCON IS ACCEPTED
2235 20 T1MS
2236 177777 -1

2237 4253 JSR STOCW ; CLEAR OUTPUT DONE BIT
2240 0 0
2241 4353 JSR OUTCK
2242 20 T1MS
2243 4 OCW. T
2244 104400 104400 ; STATUS CHANGE: NEW STATUS = 1
2245 177777 -1
```

; TEST SINGLE-CHARACTER MODE OUTPUT

2246	4205	JSR	FBAK	
2247	4235	JSR	OUTIM	
2250	4253	JSR	STOCW	; SET TO 110 BAUD, 8 BITS, NO PARITY,
2251	50360	50360		; 2 STOP BITS = .1 SEC/CHAR.
2252	4353	JSR	OUTCK	; WAIT TILL PCON ACCEPTED
2253	20	T1MS		
2254	177777	-1		
2255	4253	JSR	STOCW	; SEND A CHARACTER
2256	60240	60240		
2257	60525	DIAS	O, MUX	; CLEAR MUX STATUS WORD
2260	4353	JSR	OUTCK	; DONE BIT SHOULD COME ON AT ONCE
2261	20	T1MS		
2262	4	OCW. T		
2263	166240	166240		
2264	100000	MSW. T		; MUX STATUS WORD = OUTPUT DONE
2265	10000	010000		
2266	177777	-1		
2267	4253	JSR	STOCW	; SEND SECOND CHARACTER
2270	60125	60125		
2271	4353	JSR	OUTCK	; DONE BIT AGAIN COMES ON AT ONCE
2272	22	T2MS		
2273	4	OCW. T		
2274	166125	166125		
2275	100000	MSW. T		; MUX STATUS WORD = MULTIPLE OUTPUT
2276	10000	010000		
2277	177777	-1		
2300	4253	JSR	STOCW	; SEND THIRD CHARACTER
2301	60252	60252		
2302	4353	JSR	OUTCK	; THIS TIME DONE TAKES .1 SEC.
2303	30	T. 1SE		
2304	4	OCW. T		
2305	166252	166252		
2306	177777	-1		
2307	26137	LDA	1, @RCVR	; CHECK THAT XMTR IS CONNECTED TO RCVR
2310	125014	SKZ	1, 1	
2311	403	JMP	. +3	
2312	6012	JSR	@. FALT	
2313	3726	FNDIN		; "NO INPUT (CONN. XMTR 3 TO RCVR 2)"
2314	4364	JSR	TIMCK	; WAIT TWO CHARACTER TIMES
2315	32	T. 2SE		
2316	4	OCW. T		
2317	166252	166252		
2320	100002	MSW. T!S		; MUX STATUS WORD = R-T + IN + OUT
2321	110000	110000		
2322	177777	-1		

; TEST XMTR OUTPUT ==> RCVR INPUT, IN SINGLE CHARACTER MODE

2323	4205	JSR	FBAK	
2324	4235	JSR	OUTIM	
2325	6014	JSR	@.S960	; SET XMTR AND RCVR TO 9600 BAUD
2326	4261	JSR	STOCU	; SEND A CHARACTER
2327	66360	DCWS1:	66360	
2330	4353	JSR	OUTCK	
2331	20	TIMS		
2332	4	OCW. T		
2333	166360		166360	
2334	1	ICW. R		; RCVR ICW STILL = 0
2335	0	0		
2336	177777	-1		
2337	4360	JSR	INCK	
2340	20	TIMS		
2341	4	OCW. T		
2342	166360		166360	
2343	1	ICW. R		
2344	100360		100360	
2345	100002	MSW. T!S		; CHECK MUX STATUS WORD STACK (FIFO)
2346	110000		110000	; MSW = R-T + OUTPUT DONE
2347	100001	MSW. R		
2350	20000	020000		; MSW = INPUT DONE (DON'T POP STACK)
2351	100003	MSW. R!S		
2352	20000	020000		; MSW = SAME, NOW POP IT
2353	100001	MSW. R		
2354	0	000000		; NOW MSW = CLEAR
2355	177777	-1		
2356	102400	SUB	0,0	; CLEAR ICW
2357	42137	STA	0,@RCVR	
2360	4364	JSR	TIMCK	; CHECK THAT NO FURTHER INPUT COMES IN
2361	32	T. 2SE		
2362	1	ICW. R		
2363	0	0		
2364	100001	MSW. R		
2365	100000		100000	
2366	177777	-1		

or R-terminated here.

*100 200
Mux timed out
100 324*

; BUFGON TEST - CHECKS BUFFER POINTER COMPARISON LOGIC
; EACH TEST OBP = LOB-2, AND LOB TAKES ON THE VALUES:

;	1111111111111110	
;	0000000000000010	
;	1111111111111101	TWO PASSES ARE MADE
;	0000000000000100	FOR EACH TEST CASE
;	1111111111111011	CHECKING THAT MUX
;	0000000000001000	DOES NOT STOP WHEN
;		OBP < LOB, BUT DOES
;	1101111111111111	STOP WHEN OBP = LOB
;	0100000000000000	AND WHEN OBP > LOB
;	1011111111111111	
;	1000000000000000	

```

2367 4205 JSR FBAK
2370 6014 JSR @.S960 ; SET TO 9600 BAUD
2371 102120 ADCZL 0,0
2372 41007 BQ1: STA 0,LOB.,2 ; SET UP OBP AND LOB
2373 126120 ADCZL 1,1
2374 123000 ADD 1,0
2375 41005 STA 0,OBP.,2
2376 102000 ADC 0,0
2377 40116 STA 0,COUNT
2400 4253 BQ2: JSR STOCW ; START AUTO OUTPUT
2401 20000 20000
2402 4353 JSR OUTCK
2403 22 T2MS
2404 177777 -1
2405 21007 LDA 0,LOB.,2
2406 25005 LDA 1,OBP.,2
2407 10116 ISZ COUNT ; SECOND PASS ?
2410 101400 INC 0,0 ; YES, OBP SHOULD = LOB+1
2411 6011 TEST
2412 3776 FMCWD ; OBP IS WRONG
2413 20116 LDA 0,COUNT
2414 101015 SNZ 0,0 ; FIRST PASS ?
2415 763 JMP BQ2 ; YES
2416 35007 LDA 3,LOB.,2
2417 160122 COMZL 3,0,SZC ; MODIFY LOB FOR NEXT TEST
2420 101240 MOVOR 0,0
2421 175114 MOVL# 3,3,SZR ; LAST CASE (LOB = 100000) DONE ?
2422 750 JMP BQ1 ; NO
2423 4205 JSR FBAK
2424 40123 INLST: STA 0,INISW ; SET "INITIAL SWITCH"
2425 571 JMP MODPC

```

```

; MAIN TEST LOOP; XMTR OUTPUT ==> RCVR INPUT, AUTOMATIC BUFFER MODE
; (EXCEPT VERY FIRST TEST WHICH IS AUTO OUT AND SINGLE IN)
; THIS PART IS TABLE-DRIVEN BY THE CONTROL WORDS STARTING AT "LIST"

```

```

2426 4205 TLOOP: JSR FBAK
2427 40120 STA 0,TS
2430 4221 JSR EITTY
2431 34115 LDA 3,PNTR ; SEND NEW CONTROL WORDS TO PORTS
2432 21400 LDA 0,0,3
2433 25401 LDA 1,1,3
2434 34137 LDA 3,RCVR
2435 41001 STA 0,OCW.,2
2436 45401 STA 1,OCW.,3
2437 63025 DOC 0,MUX ; TURN MUX ON
2440 61025 DOA 0,MUX ; PROD MUX FOR IMMEDIATE ACTION
2441 21001 LDA 0,OCW.,2 ; WAIT FOR MUX TO ACCEPT BOTH PCW'S
2442 25401 LDA 1,OCW.,3
2443 107512 ANDL# 0,1,SZC
2444 405 JMP .+5
2445 10120 ISZ TS ; TIME-OUT ?
2446 773 JMP .-5 ; NO
2447 6012 JSR @.FALT ; YES
2450 3746 FTIMO

```

```
2451 62025      DOB      0, MUX      ; SET UP CTRL WDS FOR AUTO
2452 20140      LDA      0, IBP
2453 24135      LDA      1, DELTA
2454 34106      LDA      3, CMD
2455 175113     SSN      3, 3      ; IS CMD = REPEAT ?
2456 123000     ADD      1, 0      ; NO, STEP INPUT BUFFER THRU CORE
2457 40140      STA      0, IBP
2460 101620     INCZR   0, 0
2461 40141      STA      0, BFR
2462 34115      LDA      3, PNTR
2463 20156      LDA      0, C2
2464 163140     ADDOL   3, 0
2465 41005      STA      0, DBP. , 2
2466 24041      LDA      1, C4
2467 123000     ADD      1, 0
2470 41007      STA      0, LOB. , 2
2471 21402      LDA      0, 2, 3
2472 41001      STA      0, DCW. , 2
2473 30217      LDA      2, CM3
2474 143700     ANDS    2, 0
2475 30137      LDA      2, RCVR
2476 41000      STA      0, ICW. , 2
2477 20140      LDA      0, IBP
2500 41004      STA      0, IBP. , 2
2501 123000     ADD      1, 0
2502 41006      STA      0, LIB. , 2
2503 63325     DCP     0, MUX      ; START MUX
2504 61025     DCA     0, MUX
2505 21402      LDA      0, 2, 3
2506 101203     MOVR   0, 0, SNC   ; IS SPECIAL INSERT REQUIRED ?
2507 432        JMP     WTIND      ; NO
2510 101202     MOVR   0, 0, SZC   ; BUFFER OVERFILL TEST ?
2511 45006     STA      1, LIB. , 2 ; YES
2512 4353      JSR     OUTCK    ; WAIT TILL OUTPUT DONE
2513 36        T. 5SE
2514 177777     -1

2515 34115     LDA      3, PNTR   ; CK FLAG FOR WHICH TEST
2516 21402     LDA      0, 2, 3
2517 101200     MOVR   0, 0
2520 101202     MOVR   0, 0, SZC
2521 420        JMP     WTIND    ; VERY FIRST OR BUFFER OVERFILL TEST
2522 101202     MOVR   0, 0, SZC
2523 407        JMP     NECHO
2524 62125     DOBS   0, MUX   ; DATA CHANNEL OVERLOAD TEST
2525 4364      JSR     TIMCK
2526 34        T. 4SE
2527 177777     -1
2530 63025     DCP     0, MUX
2531 410        JMP     WTIND

2532 30137     NECHO: LDA    2, RCVR   ; NO-ECHO TEST
2533 4253      JSR     STOCW
2534 60101     60101
2535 21001     LDA      0, DCW. , 2
2536 101113     SSN    0, 0
2537 776      JMP     -2
2540 45001     STA      1, DCW. , 2
```


; WAIT FOR INPUT DONE

```

2541 24106 WTIND: LDA 1, CMD ; IF COMMAND IS NOT REPEAT,
2542 102400 SUB 0, 0 ; TIME-OUT COUNT = MAX.
2543 125112 SSP 1, 1 ; IF COMMAND IS REPEAT,
2544 60477 READS 0 ; READ TIME-OUT COUNT FROM SWITCHES
2545 101140 MOVOL 0, 0 ; COUNT-DOWN INCL. CARRY FOR 17-BITS
2546 30137 LDA 2, RCVR
2547 26137 LDA 1, @RCVR
2550 125112 SSP 1, 1 ; INPUT DONE ?
2551 404 JMP . +4 ; YES
2552 101405 INC 0, 0, SNR
2553 101002 MOV 0, 0, SZC ; TIME UP ?
2554 772 JMP . -6 ; NO

2555 20106 LDA 0, CMD
2556 101112 SSP 0, 0 ; IS CMD = REPEAT ?
2557 647 JMP TLOOP ; YES
2560 125112 SSP 1, 1 ; NO, WAS TIME UP ?
2561 403 JMP . +3
2562 6012 JSR @. FALT ; YES -- TIMED OUT
2563 103746 @FTIMO
2564 30115 LDA 2, PNTR
2565 26137 LDA 1, @RCVR ; CHECK ICW
2566 21006 LDA 0, 6, 2
2567 6011 TEST
2570 104042 @FICW ; ICW WRONG
2571 26141 LDA 1, @BFR ; CHECK FIRST 2 BYTES IN INPUT BUFFER
2572 21005 LDA 0, 5, 2
2573 6011 TEST
2574 104047 @FWDAT ; WRONG OR NO DATA IN INPUT BUFFER
2575 30137 LDA 2, RCVR ; CHECK FOR CROSSTALK INTO OTHER PORTS
2576 6525 XTALK: JSR @. NXTP
2577 34137 LDA 3, RCVR
2600 156415 SNE 2, 3
2601 406 JMP . +6
2602 25000 LDA 1, 0, 2
2603 102400 SUB 0, 0
2604 6011 TEST
2605 3526 FGARB ; GARBAGE COMING IN
2606 770 JMP XTALK

2607 4205 JSR FBAK
2610 34115 TNEXT: LDA 3, PNTR ; SET UP FOR NEXT TEST
2611 20045 LDA 0, C10
2612 117000 ADD 0, 3
2613 54115 STA 3, PNTR
2614 14116 DSZ COUNT ; LAST TEST AT THIS FREQ. & CHAR. L. ?
2615 611 JMP TLOOP ; NO

```

; PCW = PORT CONTROL WORD

```

;-----;-----;-----;-----;-----;-----;-----;-----;
; PCON ; DEV; ; ; ; PAR; TWO; NO. ; EVN; ;
; 0 ; OUTPUT ; CON; 0 ; 0 ; 0 ; 0 ; INH; STP; DATA ; PAR; F R E Q ;
; ; 1 ; 0 ; 1 ; ; ; ; ; BAU; BITS ; ; ;
;-----;-----;-----;-----;-----;-----;-----;-----;
; 0 ; 5 ; 0 ; ; ; ; ; ; ; ; ;

```

```

2616 20510 MODPC: LDA 0, NLIST ; MOD ALL PCW IN LIST FOR NXT TEST
2617 40116 STA 0, COUNT
2620 20154 LDA 0, .LIST
2621 40115 STA 0, PNTR
2622 102120 ADCZL 0, 0
2623 40124 STA 0, SWICH
2624 22115 MODCL: LDA 0, @PNTR ; MOD CHAR LENGTH IN PCW'S
2625 24054 LDA 1, C60 ; MASK FOR CHAR LENGTH (CL)
2626 134400 NEG 1, 3 ; A3 = -60 = -3 IN CL
2627 107404 AND 0, 1, SZR ; A1 = CL
2630 34050 LDA 3, C20 ; A3 = 20 = 1 IN CL
2631 162400 SUB 3, 0 ; DECR CHAR L (MOD 100)
2632 125014 SKZ 1, 1 ; IS CL = 0 ?
2633 406 JMP .+6 ; NO
2634 24044 LDA 1, C7 ; YES; MOD FREQ IN EACH PCW
2635 134400 NEG 1, 3 ; A3 = -7
2636 107404 AND 0, 1, SZR ; A1 = FREQ
2637 176520 SUBZL 3, 3 ; A3 = 1
2640 162400 SUB 3, 0 ; DECR FREQ (MOD 10)
2641 24123 LDA 1, INISW ; IS INITIAL FLAG SET ?
2642 125134 MOVZL# 1, 1, SZR
2643 405 JMP .+5 ; NO
2644 30055 LDA 2, C67 ; YES, SET CL = 3 & F = 7
2645 154000 COM 2, 3
2646 163400 AND 3, 0
2647 143000 ADD 2, 0
2650 42115 STA 0, @PNTR ; STORE MODIFIED PCW
2651 10115 ISZ PNTR
2652 10124 ISZ SWICH
2653 751 JMP MODCL ; MOD SECOND PCW IN SET
2654 34115 LDA 3, PNTR ; GO TO FIRST PCW OF NEXT SET
2655 30043 LDA 2, C6
2656 157000 ADD 2, 3
2657 54115 STA 3, PNTR
2660 14116 DSZ COUNT ; LAST SET DONE ?
2661 741 JMP MODCL-2 ; NO
2662 125015 SNZ 1, 1 ; IS INITIAL FLAG SET ?
2663 16441 DSZ @.LSTO ; YES; MOD F FOR HALF FREQ TEST
2664 125112 SSP 1, 1 ; INITIALIZATION FOR PART 1 ?
2665 2005 JMP @.ST2 ; YES - RETURN TO STRT2
2666 125014 SKZ 1, 1 ; VERY FIRST AUTO BUFFER TEST ?
2667 405 JMP .+5 ; NO
2670 30043 LDA 2, C6 ; PICK UP COUNT
2671 34154 LDA 3, .LIST ; AND POINTER
2672 10123 ISZ INISW ; CLR INITIAL FLAG
2673 424 JMP STORE

```

```
2674 24055 LDA 1,C67
2675 107405 AND 0,1,SNR ; IS CL = 0 & F = 0 ?
2676 452 JMP SPCHA ; YES, MAIN TEST LOOP DONE
```

```
; SET UP COUNT AND PNTR FOR NEXT TESTS
; COUNT (A2) - NO. OF TESTS TO BE DONE FROM "LIST"
; PNTR (A3) - STARTING ADDRESS OF FIRST TEST
```

```
2677 30041 LDA 2,C4 ; SET UP FOR NORMAL TEST
2700 34425 LDA 3,.LST1 ; (PNTR = LIST1, COUNT = 4)
2701 24054 LDA 1,C60
2702 107415 AND# 0,1,SNR ; IF CL = 0 (5-BIT), SET COUNT = 2
2703 151220 MOVZR 2,2 ; (BEC. 1602 GIVES 1.5 STOP BAUDS)
2704 24053 LDA 1,C40
2705 107625 ANDZR 0,1,SNR ; IS CL = 2 OR 3 ?
2706 411 JMP STORE ; NO
2707 107405 AND 0,1,SNR ; IS CL = 2 (7-BIT) ?
2710 406 JMP L7TST ; YES
2711 24043 LDA 1,C6 ; NO, CL = 3 (8-BIT)
2712 107405 AND 0,1,SNR ; IS F = 0 OR 1 ?
2713 404 JMP STORE ; YES
2714 34410 LDA 3,.LST0 ; NO, DO HALF FREQ TEST
2715 151401 INC 2,2,SKP ; (PNTR = LIST0, COUNT = 5)
2716 30411 L7TST: LDA 2,N7LST ; DO 7-BIT CHAR. L. TESTS
2717 50116 STORE: STA 2,COUNT
2720 54115 STA 3,PNTR
2721 2401 GO
2722 2426 TLOOP
```

```
2723 3217 .NXTP: NEXTP
2724 3263 .LST0: LIST0
2725 3273 .LST1: LIST1
2726 12 NLIST: 12 ; NUMBER OF TESTS IN LIST
2727 10 N7LST: 10 ; NUMBER OF TESTS FOR CL = 7
2730 177740 CM40: -40
2731 1002 C1002: 1002
2732 50000 C50K: 50000
```

```
; SUBROUTINE TO SET XMTR AND RCVR TO 9600 BAUD
```

```
2733 54125 SET96: STA 3,TSR ; SET XMTR AND RCVR TO 9600 BAUD
2734 30136 LDA 2, XMTR
2735 4253 JSR STDCW
2736 50377 50377
2737 30137 LDA 2, RCVR
2740 4253 JSR STDCW
2741 40377 40377
2742 4364 JSR TIMCK
2743 20 T1MS
2744 177777 -1
2745 60125 NIOS MUX ; CLEAR MUX INTERRUPT STACK
2746 60125 NIOS MUX
2747 2125 JMP @TSR
```

; TEST SPECIAL CHARACTER RECOGNITION LOGIC USING PROM

```
2750 4205 SPCHA: JSR   FBAK
2751 4762         JSR   SET96
2752 34100        LDA   3, BOTWD
2753 30136        LDA   2, XMTR   ; SET UP A 257 BYTE OUTPUT BUFFER
2754 161140       MOVOL 3, 0
2755 41005        STA   0, OBP., 2
2756 24753        LDA   1, C1002 ; FILL IT WITH ALL 256 ASCII CHAR. S
2757 102520       SUBZL 0, 0
2760 175400       INC   3, 3
2761 41400        STA   0, 0, 3
2762 123023       ADDZ   1, 0, SNC
2763 775          JMP   . -3
2764 24447        LDA   1, C. A.
2765 45401        STA   1, 1, 3   ; PUT AN "A" IN 257TH BYTE
2766 161520       INCZL 3, 0
2767 41007        STA   0, LOB., 2
2770 34137        LDA   3, RCVR   ; SET UP A 257 BYTE INPUT BUFFER
2771 101400       INC   0, 0
2772 41404        STA   0, IBP., 3
2773 24166        LDA   1, C401
2774 123000       ADD   1, 0
2775 41406        STA   0, LIB., 3
2776 115220       MOVZR 0, 3
2777 24435        LDA   1, C. : X
3000 45400        STA   1, 0, 3   ; PUT AN "X" IN 258TH BYTE
3001 30427        LDA   2, . POUT ; CLEAR SPECIAL CHAR. BIT MAP AREA
3002 102400       SUB   0, 0
3003 24725        LDA   1, CM40
3004 41000        STA   0, 0, 2
3005 151400       INC   2, 2
3006 125404       INC   1, 1, SZR
3007 775          JMP   . -3
3010 20722        LDA   0, C50K   ; START AUTO INPUT W. SPL. INTPT. REQ.
3011 42137        STA   0, @RCVR
3012 30136 SPCH1: LDA   2, XMTR   ; START AUTO OUTPUT W. SPL. INTPT. REQ.
3013 4253         JSR   STOCW
3014 30101        30101
3015 30136 OUTST: LDA   2, XMTR
3016 21001        LDA   0, OCW., 2 ; TEST FOR OUTPUT DONE BIT
3017 101113       SSN   0, 0   ; OUTPUT DONE ?
3020 416          JMP   INTST   ; NO
3021 25005        LDA   1, OBP., 2
3022 35007        LDA   3, LOB., 2
3023 136033       SLS   1, 3   ; IS OUTPUT BUFFER EMPTY ?
3024 412          JMP   INTST   ; YES
3025 30403        LDA   2, . POUT ; POST BIT IN OUTPUT BIT MAP
3026 4451         JSR   STBIT
3027 763          JMP   SPCH1   ; TURN AUTO OUTPUT BACK ON

3030 12000 . POUT: 12000 ; OUTPUT BIT MAP
3031 12020 . PIN: 12020  ; INPUT BIT MAP
3032 4740 . PROM: PROM   ; STANDARD BIT MAPS (BOTH OUT & IN)
3033 40400 C. A. : "A*L
3034 130 C. X: "X
3035 140530 C. AX: @"A*L+"X
```

```
3036 22137 INTST: LDA 0, @RCVR ; TEST FOR INPUT DONE
3037 101113 SSN 0, 0 ; INPUT DONE ?
3040 755 JMP OUTST ; NO
3041 34137 LDA 3, RCVR
3042 11404 ISZ IBP, 3
3043 25404 LDA 1, IBP, 3
3044 31406 LDA 2, LIB, 3
3045 132033 SLS 1, 2 ; IS INPUT BUFFER FULL ?
3046 406 JMP COMPA ; YES
3047 30762 LDA 2, .PIN ; POST BIT IN INPUT BIT MAP
3050 4427 JSR STBIT
3051 20661 LDA 0, C50K ; TURN AUTO INPUT BACK ON
3052 42137 STA 0, @RCVR
3053 742 JMP OUTST

3054 151220 COMPA: MOVZR 2, 2 ; CHECK NEXT BYTE AFTER IN BUFFER
3055 25000 LDA 1, 0, 2
3056 20757 LDA 0, C. AX
3057 6011 TEST ; IS IT STILL AN "X" ?
3060 4116 FBOVF ; NO, BUFFER OVERFILL
3061 20053 LDA 0, C40 ; COMPARE BIT MAP AGAINST STANDARD
3062 40116 STA 0, COUNT
3063 20745 LDA 0, . POUT
3064 40115 STA 0, PNTR
3065 30745 LDA 2, . PROM
3066 21000 COMP1: LDA 0, 0, 2
3067 26115 LDA 1, @PNTR
3070 6011 TEST
3071 4035 FSPCH ; SPECIAL CHARACTER RECOGNITION ERROR
3072 151400 INC 2, 2
3073 10115 ISZ PNTR
3074 14116 DSZ COUNT
3075 771 JMP COMP1
3076 425 JMP TMSW

3077 24377 STBIT: LDA 1, C377 ; SET BIT CORRESP. TO SPL. CHAR.
3100 123400 AND 1, 0
3101 126520 SUBZL 1, 1
3102 101222 MOVZR 0, 0, SZC
3103 125120 MOVZL 1, 1
3104 101222 MOVZR 0, 0, SZC
3105 127120 ADDZL 1, 1
3106 101223 MOVZR 0, 0, SNC
3107 403 JMP . +3
3110 127120 ADDZL 1, 1
3111 127120 ADDZL 1, 1
3112 101222 MOVZR 0, 0, SZC
3113 125300 MOVS 1, 1
3114 113000 ADD 0, 2
3115 21000 LDA 0, 0, 2
3116 123000 ADD 1, 0
3117 107415 AND# 0, 1, SNR
3120 122400 SUB 1, 0
3121 41000 STA 0, 0, 2
3122 1400 JMP 0, 3
```

; TEST MUX STATUS WORD

```
3123 4205 TMSW: JSR FBAK
3124 102000 ADC 0,0
3125 40124 STA 0,SWICH
3126 4605 JSR SET96
3127 20435 LDA 0,D19 ; FIRST PASS WE XMIT & RCV 19 CHARS.
3130 40116 MSW0: STA 0,COUNT
3131 4253 JSR STOCW
3132 60101 DCWSI: 60101
3133 102400 SUB 0,0
3134 42137 STA 0,@RCVR
3135 4360 JSR INCK
3136 22 T2MS
3137 177777 -1
3140 14116 DSZ COUNT
3141 770 JMP MSW0+1
3142 10124 ISZ SWICH ; FIRST PASS ?
3143 423 JMP MSW2 ; NO
3144 20420 LDA 0,D19 ; CHECK THE STATUS WORD STACK
3145 40116 STA 0,COUNT
3146 4362 MSW1: JSR CHECK ; THERE SHOULD BE 19 OUTPUT DONES
3147 100002 MSW. T!S
3150 10000 10000
3151 100003 MSW. R!S ; ALTERNATING WITH 19 INPUT DONES
3152 20000 20000
3153 177777 -1
3154 14116 DSZ COUNT ; HAVE WE CONFIRMED ALL 19 PAIRS ?
3155 771 JMP MSW1 ; NO
3156 4362 JSR CHECK ; CHECK THAT STACK IS NOW EMPTY
3157 100001 MSW. R
3160 0 0
3161 177777 -1
3162 20403 LDA 0,D20 ; PREPARE FOR SECOND PASS (20 CHAR.)
3163 745 JMP MSW0

3164 23 D19: 23
3165 24 D20: 24

3166 4362 MSW2: JSR CHECK ; CK. STATUS WORD STACK AFTER 2D PASS
3167 100004 MSW.
3170 40000 40000 ; STACK OVERFLOW
3171 100004 MSW.
3172 0 0
3173 177777 -1
```

```

3174 4205 JSR FBAK ; END OF PART 2 OF DIAGNOSTIC
3175 20106 PTDON: LDA 0, CMD
3176 103112 ADDL# 0, 0, SZC ; IS CMD = F ?
3177 2406 JMP @. IN. C ; YES, CONTINUE FROM NEXT "FBAK"
3200 6013 JSR @. ENTP ; NO, ENABLE TYPE-OUT
3201 4324 JSR TYPE
3202 164016 @"[-H*F+"A-H*F+"O-H ; " AOK"
3203 45400 "K*L
3204 2006 JMP @. WAIT

3205 1712 . IN. C: IN. C

3206 30136 IN. N: LDA 2, XMTR ; GO TO NEXT PORT PAIR
3207 4410 JSR NEXTP
3210 50136 STA 2, XMTR
3211 30137 LDA 2, RCVR
3212 4405 JSR NEXTP
3213 50137 STA 2, RCVR
3214 4205 JSR FBAK ; CONT. PT. IN "F" FOR ASYNC. PORTS
3215 2401 GO
3216 2057 PART2

3217 24133 NEXTP: LDA 1, BLKL ; ADD BLKL TO PORT ADDRESS
3220 133000 ADD 1, 2
3221 24134 LDA 1, LPORT
3222 146433 SLE 2, 1 ; IS NEW PORT ADDRESS > LAST PORT ?
3223 30132 LDA 2, BASE ; YES, CHANGE IT TO BASE
3224 1400 JMP 0, 3

```

; INTERRUPT SERVICE ROUTINES

```

3225 65477 ISMUX: INTA 1 ; ONLY MUX WAS MASKED IN
3226 125014 SKZ 1, 1
3227 403 JMP . +3
3230 6012 JSR @. FALT
3231 3702 FINTP ; NO INTERRUPT ACKNOWLEDGEMENT
3232 20051 LDA 0, CMUX
3233 106415 SNE 0, 1
3234 63625 SKPDN MUX ; CONFIRM MUX DONE FLAG
3235 413 JMP IFALT
3236 60525 DIAS 0, MUX ; READ IN MUX STATUS WORD & CLEAR IT
3237 151400 INC 2, 2
3240 60177 INTEN
3241 2000 JMP @0

3242 65477 ISTTY: INTA 1 ; ONLY TTI WAS MASKED IN
3243 20216 LDA 0, CTTI
3244 106415 SNE 0, 1
3245 63610 SKPDN TTI
3246 402 JMP IFALT
3247 2006 JMP @. WAIT
3250 6012 IFALT: JSR @. FALT
3251 3624 FINTP ; WRONG DEVICE CODE OR DONE FLAG

3252 50520 CKS5: 50520 ; CHECKSUM

```

; TABLE OF CONTROL WORDS DRIVING MAIN IN-OUT TEST LOOP

; TEST CONTROL WORD (LEFT HALVES OF DESIRED OCW AND ICW):

26100 G= 40000/L+26000 ;NORMAL AUTO OUT AND AUTO IN
26140 GAE= 60000/L+26000 ;AUTO OUT AND AUTO ECHO IN
26120 G.7= 50000/L+26000 ;AUTO IN WITH 7-BIT ASCII MODE

; VERY FIRST AUTO BUFFER TEST

3253 56377 LIST: 56377 ;XMTR PCON = 8 BIT, NO PAR., 2 STOP
3254 40377 40377 ;RCVR PCON = SAME
3255 26003 26003 ;OUTMO = AUTO, INMO = SINGLE
3256 402 001*L+002;OUTGOING CHAR. 1 AND 2
3257 1404 003*L+004;OUTGOING CHAR. 3 AND 4
3260 0 0 ;INPUT BUFFER SHOULD BE EMPTY
3261 100402 100402 ;ICW SHD BE CHAR 2 W MUX SVC OVLD
3262 1 EAUTO ;POINTER TO TEST NAME FOR TYPE-OUT IF ERROR

; HALF-FREQUENCY TEST IS NOT DONE AT 150 OR 110 BAUD, & ONLY FOR 8-BIT CH.

3263 56276 LISTO: 56276 ;OUT=8 BIT, NO PAR, 1 STOP
3264 40267 40267 ;IN=SAME, FOR HALF FREQ. TEST
3265 26100 G ;NORMAL TEST CONTROL WORD
3266 105517 213*L+117;OUT CHAR 1 & 2 (BECOMES IN CHAR 1-4)
3267 125252 252*L+252;OUT CHAR 3 & 4 (NOT RGD THIS TEST)
3270 117340 236*L+340;IN CHAR 1 & 2 SHD BE THIS
3271 140230 140230 ;ICW SHD BE CHAR 4 W NO FLAGS
3272 5 EHFRQ ;HALF-FREQUENCY TEST

; THE NEXT TWO TESTS ARE DONE AT ALL FREQUENCIES AND ALL CHAR. LENGTHS

3273 56077 LIST1: 56077 ;OUT = EVEN PAR, 1 STOP
3274 40277 40277 ;IN = NO PAR, 1 STOP
3275 26100 G
3276 404 001*L+004;OUT 1 & 2
3277 252 000*L+252;OUT 3 & 4
3300 404 001*L+004;IN CHAR 1 & 2
3301 146000 146000 ;ICW FLAG = NO STOP (BREAK)
3302 5015 EBRK*L+EPARE ;EVEN PARITY/BREAK

3303 56067 56067 ;OUT = ODD PAR, 1 STOP
3304 40267 40267 ;IN = NO PAR, 1 STOP
3305 26100 G
3306 17011 036*L+011
3307 14652 031*L+252
3310 17011 036*L+011
3311 146031 146031 ;ICW FLAG = NO STOP (BREAK)
3312 5021 EBRK*L+EPARO ;ODD PARITY/BREAK

; THE NEXT TWO TESTS ARE NOT DONE FOR 5-BIT CHAR. LENGTH
; (BECAUSE THE 1602 UART GIVES 1.5 STOP BAUDS)

3313	56367	56367	;OUT=NO PAR, 2 STOP
3314	40077	40077	;IN=EVEN PAR, 1 STOP
3315	26100	Q	
3316	15015	032*L+015	
3317	10652	021*L+252	
3320	15015	032*L+015	
3321	142021	142021	;ICW SHD BE CHAR 3 W PAR ERR
3322	15	EPARE	;EVEN PARITY

3323	56377	56377	;OUT = NO PAR, 2 STOP
3324	40067	40067	;IN = ODD PAR, 1 STOP
3325	26100	Q	
3326	7422	017*L+022	
3327	17652	037*L+252	
3330	7422	017*L+022	
3331	142037	142037	;ICW FLAG = PAR ERROR
3332	21	EPARO	;ODD PARITY

; THE REMAINING TESTS ARE DONE FOR 7-BIT CHAR. LENGTH ONLY

3333	56157	56157	;OUT = EVEN PAR, 2 STOP
3334	40157	40157	;IN = EVEN PAR, 2 STOP
3335	26120	Q.7	;ICW = AUTO WITH 7-BIT ASCII MODE
3336	40502	101*L+102	
3337	41704	103*L+304	
3340	140702	301*L+302	
3341	150304	150304	;ICW = CHAR 4 WITH NO FLAGS
3342	25	E7BIT	;7-BIT ASCII

3343	56157	56157	;OUT = EVEN PAR, 2 STOP
3344	40257	40257	;IN = NO PAR, 1 STOP
3345	26145	GAE+5	;INSERT SPECIAL PGM FOR NO-ECHO TEST
3346	104133	210*L+133	
3347	3407	007*L+007	
3350	4133	010*L+133	
3351	161007	161007	;ICW = CHAR 3 WITH NO ECHO FLAG
3352	32	ENEKO	;NO-ECHO

3353	56347	56347	;OUT = NO PAR, 2 STOP
3354	40147	40147	;IN = ODD PAR, 2 STOP
3355	26101	Q+1	;INSERT SPL PGM FOR DATA CHANNEL OVERLOAD TEST
3356	57731	137*L+331	
3357	42505	105*L+105	
3360	57531	137*L+131	
3361	144105	144105	;ICW = CHAR 4 W DCH-OVLD FLAG
3362	35	EDCOV	;DATA CHANNEL OVERLOAD

3363	56157	56157	;OUT = EVEN PAR, 2 STOP
3364	40157	40157	;IN = SAME
3365	26103	Q+3	;INSERT SPL PGM FOR BUFFER OVERFILL TEST
3366	13137	026*L+137	
3367	125252	252*L+252	
3370	13000	026*L+000	
3371	140537	140537	;ICW = CHAR 2 WITH MUX SVC OVERLOAD
3372	43	EBOVF	;BUFFER OVERFILL

3373	72647	CKS6: 72647	;CHECKSUM FOR ABOVE LIST
------	-------	-------------	--------------------------

; "TEST" COMPARES A0 WITH A1 AND SKIP RETURNS IF EQUAL, ELSE FALLS INTO:
; "FAULT" TYPES THE CHARACTER STRING POINTED TO BY THE WORD FOLLOWING
; THE JSR AND RETURNS TO "WAIT", UNLESS THE FAULT IS IN MAIN TEST LOOP
; IN WHICH CASE IT TYPES OUT ADDITIONAL INFO

```
3374 106415 TEST. : SNE      0, 1
3375   1401          JMP      1, 3
3376  54113 FAULT: STA      3, A3
3377  40110          STA      0, A0
3400  44111          STA      1, A1
3401  50112          STA      2, A2
3402  20106          LDA      0, CMD
3403 101112          SSP      0, 0      ; IS CMD = R ?
3404   205           JMP      FBAK      ; YES, REPEAT FROM "FALLBACK"
3405   6013          JSR      @.ENTP    ; NO, ENABLE TYPE-OUT
3406 102520          SUBZL    0, 0
3407  40107          STA      0, SPFLQ
3410   4324          JSR      TYPE      ; 2 BELLS, CR, LF, "FAULT: "
3411   3407          7*L+7
3412 175645 @"_-H*F+"^-H*F+"F-H
3413 101213 @"A-H*F+"U-H*F+"L-H
3414  52072 "T*L+":
3415  20000 " *L
3416  32113          LDA      2, @A3
3417 151100          MOVL     2, 2
3420 151220          MOVZR   2, 2
3421   4326          JSR      TPMSG     ; TYPE MESSAGE
3422  22113          LDA      0, @A3
3423 101113          SSN      0, 0      ; IS ERROR IN MAIN TEST LOOP ?
3424   2006          JMP      @.WAIT    ; NO, WAIT FOR NEXT COMMAND
3425   4324          JSR      TYPE      ; YES, TYPE " IN "
3426 164415 @"[-H*F+"I-H*F+"N-H
3427  20000 " *L
3430  30115          LDA      2, PNTR    ; PICK UP POINTER TO NAME OF TEST
3431  21007          LDA      0, 7, 2
3432  24377 ERR1: LDA      1, C377
3433 107405          AND      0, 1, SNR
3434   411           JMP      ERR11
3435 122700          SUBS     1, 0
3436  40120          STA      0, TS
3437  30405          LDA      2, .EMSG
3440 133000          ADD      1, 2
3441   4326          JSR      TPMSG
3442  20120          LDA      0, TS
3443   767           JMP      ERR1

3444   4053 .EMSG: EMSG.
```

```

3445 4324 ERR11:JSR TYPE ; " TEST AT "
3446 165144 @"-H*F+"T-H*F+"E-H
3447 145172 @"-S-H*F+"T-H*F+"[-H
3450 101172 @"-A-H*F+"T-H*F+"[-H
3451 0 0
3452 30115 LDA 2,PNTR ; DETERMINE FREQUENCY
3453 21001 LDA 0,1,2
3454 24434 LDA 1,D11
3455 30044 LDA 2,C7
3456 143405 AND 2,0,SNR ; IS BAUD RATE = 110 ?
3457 406 JMP .+6 ; YES
3460 24431 LDA 1,D15 ; NO, CALCULATE BAUD RATE/10
3461 100401 NEG 0,0,SKP
3462 127000 ADD 1,1
3463 101404 INC 0,0,SZR
3464 776 JMP .-2
3465 4352 JSR TPDEC ; TYPE BAUD RATE/10
3466 4324 JSR TYPE
3467 163501 @"-Z-H*F+"[-H*F+"B-H ; "O BAUD, CHAR.L. = "
3470 101203 @"-A-H*F+"U-H*F+"D-H
3471 26040 ",*L+"
3472 104340 @"-C-H*F+"H-H*F+"A-H
3473 51056 "R*L+"
3474 46056 "L*L+"
3475 20075 " *L+"=
3476 20000 " *L
3477 22115 LDA 0,@PNTR ; DETERMINE CHARACTER LENGTH
3500 24054 LDA 1,C60
3501 123520 ANDZL 1,0
3502 103120 ADDZL 0,0
3503 103300 ADDS 0,0
3504 24042 LDA 1,C5
3505 107000 ADD 0,1
3506 4352 JSR TPDEC ; TYPE IT
3507 2006 JMP @.WAIT

3510 13 D11: 13
3511 17 D15: 17

```

; MESSAGES FILE

; CODING OF MESSAGES: TEXT MAY BE STORED EITHER 2 OR 3 CHARACTERS/WORD.
; IF MSB=0, WORD = ANY 2 7-BIT ASCII CHAR., WITH 0 BYTE = TERMINATOR;
; IF MSB=1, WORD = 3 "ALPHA" CHARACTERS, EACH 5 BITS CODED AS FOLLOWS:

; 00-07	A	B	C	D	E	F	G	H
; 10-17	I	J	K	L	M	N	O	P
; 20-27	Q	R	S	T	U	V	W	X
; 30-37	Y	0	SPACE	TERM.	-	LF	CR	=

; THE ASSEMBLER PRODUCES THESE CODES BY SUBTRACTING 101 (=H) FROM THEIR
; ASCII VALUES. THE LAST 7 CHARACTERS ARE PRODUCED BY USING (SEE TP3CH):

; ASCII 132 = Z	==>	CODE 31 = 0
; ASCII 133 = [==>	CODE 32 = SPACE
; ASCII 134 = \	==>	CODE 33 = END STRING
; ASCII 135 =]	==>	CODE 34 = -
; ASCII 136 = ^	==>	CODE 35 = LINE FEED
; ASCII 137 = _	==>	CODE 36 = CARRIAGE RETURN
; ASCII 140 = `	==>	CODE 37 = =

101 H= 101
40 F= 40

; FAULT MESSAGES

3512 PANIC= ; PANIC!! PROGRAM BEING CLOBBERED
3512 136015 @"P-H*F+"A-H*F+"N-H
3513 44503 "I*L+"C
3514 20441 "!*L+"!
3515 164761 @"[-H*F+"P-H*F+"R-H
3516 134321 @"O-H*F+"G-H*F+"R-H
3517 100632 @"A-H*F+"M-H*F+"[-H
3520 102210 @"B-H*F+"E-H*F+"I-H
3521 132332 @"N-H*F+"G-H*F+"[-H
3522 104556 @"C-H*F+"L-H*F+"O-H
3523 102044 @"B-H*F+"B-H*F+"E-H
3524 142203 @"R-H*F+"E-H*F+"D-H
3525 0 0

3526 FGARB= ; GARBAGE COMING IN
3526 114021 @"G-H*F+"A-H*F+"R-H
3527 102006 @"B-H*F+"A-H*F+"G-H
3530 111502 @"E-H*F+"[-H*F+"C-H
3531 134610 @"O-H*F+"M-H*F+"I-H
3532 132332 @"N-H*F+"G-H*F+"[-H
3533 120673 @"I-H*F+"N-H*F+"-\-H

3534 FNOMX=. ; NO MUX ACTION
3534 132732 @"N-H*F+"O-H*F+"[-H
3535 131227 @"M-H*F+"U-H*F+"X-H
3536 164002 @"[-H*F+"A-H*F+"C-H
3537 146416 @"T-H*F+"I-H*F+"O-H
3540 47000 "N*L

3541 FBLKL=. ; CANNOT DETERMINE VALID BLOCK LENGTH
3541 104015 @"C-H*F+"A-H*F+"N-H
3542 132723 @"N-H*F+"O-H*F+"T-H
3543 164144 @"[-H*F+"D-H*F+"E-H
3544 146221 @"T-H*F+"E-H*F+"R-H
3545 130415 @"M-H*F+"I-H*F+"N-H
3546 111525 @"E-H*F+"[-H*F+"V-H
3547 100550 @"A-H*F+"L-H*F+"I-H
3550 107501 @"D-H*F+"[-H*F+"B-H
3551 126702 @"L-H*F+"O-H*F+"C-H
3552 125513 @"K-H*F+"[-H*F+"L-H
3553 110646 @"E-H*F+"N-H*F+"G-H
3554 146373 @"T-H*F+"H-H*F+"\-H

3555 FNSEQ=. ; PORTS NOT IN SEQUENCE - SWITCHES SET WRONG ?
3555 136721 @"P-H*F+"O-H*F+"R-H
3556 147132 @"T-H*F+"S-H*F+"[-H
3557 132723 @"N-H*F+"O-H*F+"T-H
3560 164415 @"[-H*F+"I-H*F+"N-H
3561 165104 @"[-H*F+"S-H*F+"E-H
3562 141204 @"Q-H*F+"U-H*F+"E-H
3563 132104 @"N-H*F+"C-H*F+"E-H
3564 165632 @"[-H*F+"]-H*F+"[-H
3565 145310 @"S-H*F+"W-H*F+"I-H
3566 146107 @"T-H*F+"C-H*F+"H-H
3567 111132 @"E-H*F+"S-H*F+"[-H
3570 144223 @"S-H*F+"E-H*F+"T-H
3571 165321 @"[-H*F+"W-H*F+"R-H
3572 134646 @"O-H*F+"N-H*F+"G-H
3573 20077 " *L+"?
3574 0 0

3575 FBUSY=. ; BUSY FLAG
3575 103222 @"B-H*F+"U-H*F+"S-H
3576 161505 @"Y-H*F+"[-H*F+"F-H
3577 126006 @"L-H*F+"A-H*F+"G-H
3600 0 0

3601 FDONE= . ; DONE FLAG: -12 V. MISSING ?
3601 106715 @"D-H*F+"D-H*F+"N-H
3602 111505 @"E-H*F+"[-H*F+"F-H
3603 126006 @"L-H*F+"A-H*F+"G-H
3604 35040 " : *L+"
3605 26461 "-*L+"1
3606 31040 "2*L+"
3607 53056 "V*L+"
3610 164610 @"[-H*F+"M-H*F+"I-H
3611 145110 @"S-H*F+"S-H*F+"I-H
3612 132332 @"N-H*F+"G-H*F+"[-H
3613 37400 "?*L

3614 FDEV= . ; MUX DEVICE CODE WRONG
3614 131227 @"M-H*F+"U-H*F+"X-H
3615 164144 @"[-H*F+"D-H*F+"E-H
3616 152402 @"V-H*F+"I-H*F+"C-H
3617 111502 @"E-H*F+"[-H*F+"C-H
3620 134144 @"O-H*F+"D-H*F+"E-H
3621 165321 @"[-H*F+"W-H*F+"R-H
3622 134646 @"O-H*F+"N-H*F+"G-H
3623 0 0

3624 FINT= . ; ILLEGAL INTERRUPT OCCURRED
3624 120553 @"I-H*F+"L-H*F+"L-H
3625 110300 @"E-H*F+"G-H*F+"A-H
3626 127510 @"L-H*F+"[-H*F+"I-H
3627 133144 @"N-H*F+"T-H*F+"E-H
3630 143064 @"R-H*F+"R-H*F+"U-H
3631 137172 @"P-H*F+"T-H*F+"[-H
3632 134102 @"O-H*F+"C-H*F+"C-H
3633 151061 @"U-H*F+"R-H*F+"R-H
3634 110173 @"E-H*F+"D-H*F+"\
-H

3635 FNINT= . ; MUX IS NOT PRODUCING INTERRUPTS
3635 131227 @"M-H*F+"U-H*F+"X-H
3636 164422 @"[-H*F+"I-H*F+"S-H
3637 164656 @"[-H*F+"N-H*F+"O-H
3640 147517 @"T-H*F+"[-H*F+"P-H
3641 142703 @"R-H*F+"O-H*F+"D-H
3642 150110 @"U-H*F+"C-H*F+"I-H
3643 132332 @"N-H*F+"G-H*F+"[-H
3644 120663 @"I-H*F+"N-H*F+"T-H
3645 111061 @"E-H*F+"R-H*F+"R-H
3646 150763 @"U-H*F+"P-H*F+"T-H
3647 51400 "S*L

3650 FNRTC= . ; NO REAL-TIME CLOCK - PLUGGED IN ALL THE WAY ?
3650 132732 @"N-H*F+"O-H*F+"[-H
3651 142200 @"R-H*F+"E-H*F+"A-H
3652 127623 @"L-H*F+"J-H*F+"T-H
3653 120604 @"I-H*F+"M-H*F+"E-H
3654 164113 @"[-H*F+"C-H*F+"L-H
3655 134112 @"O-H*F+"C-H*F+"K-H
3656 165634 @"[-H*F+"J-H*F+"J-H
3657 164624 @"[-H*F+"M-H*F+"U-H
3660 157517 @"X-H*F+"[-H*F+"P-H
3661 127206 @"L-H*F+"U-H*F+"G-H
3662 114203 @"O-H*F+"E-H*F+"D-H
3663 164415 @"[-H*F+"I-H*F+"N-H
3664 164013 @"[-H*F+"A-H*F+"L-H
3665 127523 @"L-H*F+"[-H*F+"T-H
3666 116232 @"H-H*F+"E-H*F+"[-H
3667 154030 @"W-H*F+"A-H*F+"Y-H
3670 20077 " *L+"?
3671 0 0

3672 FRTC= . ; REAL-TIME CLOCK WRONG
3672 142200 @"R-H*F+"E-H*F+"A-H
3673 127623 @"L-H*F+"J-H*F+"T-H
3674 120604 @"I-H*F+"M-H*F+"E-H
3675 164113 @"[-H*F+"C-H*F+"L-H
3676 134112 @"O-H*F+"C-H*F+"K-H
3677 165321 @"[-H*F+"W-H*F+"R-H
3700 134646 @"O-H*F+"N-H*F+"G-H
3701 0 0

3702 FINTP= . ; NO INTPT. ACK. -- IS INTP GETTING TO MUX ?
3702 132732 @"N-H*F+"O-H*F+"[-H
3703 120663 @"I-H*F+"N-H*F+"T-H
3704 50124 "P*L+"T
3705 27040 ". *L+"
3706 100112 @"A-H*F+"C-H*F+"K-H
3707 27040 ". *L+"
3710 171632 @"J-H*F+"J-H*F+"[-H
3711 121132 @"I-H*F+"S-H*F+"[-H
3712 120663 @"I-H*F+"N-H*F+"T-H
3713 50124 "P*L+"T
3714 27040 ". *L+"
3715 137050 @"P-H*F+"R-H*F+"I-H
3716 135050 @"O-H*F+"R-H*F+"I-H
3717 147432 @"T-H*F+"Y-H*F+"[-H
3720 114223 @"O-H*F+"E-H*F+"T-H
3721 146415 @"T-H*F+"I-H*F+"N-H
3722 115523 @"O-H*F+"[-H*F+"T-H
3723 135514 @"O-H*F+"[-H*F+"M-H
3724 151372 @"U-H*F+"X-H*F+"[-H
3725 37400 "?*L

3726 FNOIN= . ; NO INPUT (CONNECT XMTR PIN 3 TO RCVR PIN 2)
3726 132732 @"N-H*F+"O-H*F+"[-H
3727 120657 @"I-H*F+"N-H*F+"P-H
3730 151172 @"U-H*F+"T-H*F+"[-H
3731 24103 "(*L+"C
3732 134655 @"O-H*F+"N-H*F+"N-H
3733 110123 @"E-H*F+"C-H*F+"T-H
3734 165354 @"[-H*F+"X-H*F+"M-H
3735 147072 @"T-H*F+"R-H*F+"[-H
3736 136415 @"P-H*F+"I-H*F+"N-H
3737 20063 " *L+"3
3740 165156 @"[-H*F+"T-H*F+"O-H
3741 165042 @"[-H*F+"R-H*F+"C-H
3742 153072 @"V-H*F+"R-H*F+"[-H
3743 136415 @"P-H*F+"I-H*F+"N-H
3744 20062 " *L+"2
3745 24400 ")*L

3746 FTIMO= . ; MUX TIMED OUT
3746 131227 @"M-H*F+"U-H*F+"X-H
3747 165150 @"[-H*F+"T-H*F+"I-H
3750 130203 @"M-H*F+"E-H*F+"D-H
3751 164724 @"[-H*F+"O-H*F+"U-H
3752 52000 "T*L

3753 FMINT= . ; INSUFFICIENT ELAPSED TIME BEFORE MUX DONE
3753 120662 @"I-H*F+"N-H*F+"S-H
3754 150245 @"U-H*F+"F-H*F+"F-H
3755 120110 @"I-H*F+"C-H*F+"I-H
3756 110663 @"E-H*F+"N-H*F+"T-H
3757 164213 @"[-H*F+"E-H*F+"L-H
3760 100762 @"A-H*F+"P-H*F+"S-H
3761 110172 @"E-H*F+"D-H*F+"[-H
3762 146414 @"T-H*F+"I-H*F+"M-H
3763 111501 @"E-H*F+"[-H*F+"B-H
3764 110256 @"E-H*F+"F-H*F+"O-H
3765 142232 @"R-H*F+"E-H*F+"[-H
3766 131227 @"M-H*F+"U-H*F+"X-H
3767 164156 @"[-H*F+"D-H*F+"O-H
3770 132233 @"N-H*F+"E-H*F+"[-H

3771 FDOA= . ; DOA MUX TEST
3771 106700 @"D-H*F+"O-H*F+"A-H
3772 164624 @"[-H*F+"M-H*F+"U-H
3773 157523 @"X-H*F+"[-H*F+"T-H
3774 111123 @"E-H*F+"S-H*F+"T-H
3775 0 0

3776 FMCWD= . ; MUX CONTROL WORD
3776 131227 @"M-H*F+"U-H*F+"X-H
3777 164116 @"[-H*F+"C-H*F+"O-H
4000 133161 @"N-H*F+"T-H*F+"R-H
4001 134572 @"O-H*F+"L-H*F+"[-H
4002 154721 @"W-H*F+"O-H*F+"R-H
4003 42000 "D*L

4004 FMSW= . ; MUX STATUS WORD
4004 131227 @"M-H*F+"U-H*F+"X-H
4005 165123 @"[-H*F+"S-H*F+"T-H
4006 101164 @"A-H*F+"T-H*F+"U-H
4007 145526 @"S-H*F+"[-H*F+"W-H
4010 135043 @"O-H*F+"R-H*F+"D-H
4011 0 0

4012 FSTAT= . ; DEVICE STATUS (CONNECT XMTR PIN 8 TO XMTR 20)
4012 106225 @"D-H*F+"E-H*F+"V-H
4013 120104 @"I-H*F+"C-H*F+"E-H
4014 165123 @"[-H*F+"S-H*F+"T-H
4015 101164 @"A-H*F+"T-H*F+"U-H
4016 51440 "S*L+ "
4017 24103 "(*L+"C
4020 134655 @"O-H*F+"N-H*F+"N-H
4021 110123 @"E-H*F+"C-H*F+"T-H
4022 165354 @"[-H*F+"X-H*F+"M-H
4023 147072 @"T-H*F+"R-H*F+"[-H
4024 136415 @"P-H*F+"I-H*F+"N-H
4025 20070 " *L+"8
4026 165156 @"[-H*F+"T-H*F+"O-H
4027 165354 @"[-H*F+"X-H*F+"M-H
4030 147072 @"T-H*F+"R-H*F+"[-H
4031 136415 @"P-H*F+"I-H*F+"N-H
4032 20062 " *L+"2
4033 30051 "O*L+")
4034 0 0

4035 FSPCH= . ; SPL CHAR LOGIC
4035 144753 @"S-H*F+"P-H*F+"L-H
4036 164107 @"[-H*F+"C-H*F+"H-H
4037 101072 @"A-H*F+"R-H*F+"[-H
4040 126706 @"L-H*F+"O-H*F+"G-H
4041 120133 @"I-H*F+"C-H*F+" \-H

4042 4042 FICW: . ; WRONG ICW (IN ... TEST)
4043 155056 @"W-H*F+"R-H*F+"O-H
4044 132332 @"N-H*F+"G-H*F+"[-H
4045 120126 @"I-H*F+"C-H*F+"W-H
4046 0 0

4047 4047 FWDAT: . ; WRONG DATA (IN ... TEST)
4050 155056 @"W-H*F+"R-H*F+"O-H
4051 132332 @"N-H*F+"G-H*F+"[-H
4052 106023 @"D-H*F+"A-H*F+"T-H
4053 40400 "A*L

; MESSAGES DESCRIBING TESTS IN MAIN LOOP

4053 EMSG. =. -1

1 EAUTO= . -EMSG. ; AUTO BUFFER
4054 101223 @"A-H*F+"U-H*F+"T-H
4055 135501 @"O-H*F+"[-H*F+"B-H
4056 150245 @"U-H*F+"F-H*F+"F-H
4057 111073 @"E-H*F+"R-H*F+" \-H

5 EHFRQ=. -EMSG.
4060 116013 @"H-H*F+"A-H*F+"L-H
4061 113605 @"F-H*F+"J-H*F+"F-H
4062 142220 @"R-H*F+"E-H*F+"G-H
4063 150215 @"U-H*F+"E-H*F+"N-H
4064 105433 @"C-H*F+"Y-H*F+"\-H

; HALF-FREQUENCY

12 EBRK= . -EMSG.
4065 27502 "/*L+"B
4066 142200 @"R-H*F+"E-H*F+"A-H
4067 45400 "K*L

; /BREAK

15 EPARE=. -EMSG.
4070 111244 @"E-H*F+"V-H*F+"E-H
4071 133517 @"N-H*F+"[-H*F+"P-H
4072 40522 "A*L+"R
4073 27000 ". *L

; EVEN PAR.

21 EPARD=. -EMSG.
4074 134143 @"O-H*F+"D-H*F+"D-H
4075 164740 @"[-H*F+"P-H*F+"A-H
4076 51056 "R*L+".
4077 0 0

; ODD PAR.

25 E7BIT=. -EMSG.
4100 33455 "7*L+"-
4101 102423 @"B-H*F+"I-H*F+"T-H
4102 164022 @"[-H*F+"A-H*F+"S-H
4103 104410 @"C-H*F+"I-H*F+"I-H
4104 0 0

; 7-BIT ASCII

32 ENEKO=. -EMSG.
4105 132734 @"N-H*F+"O-H*F+"J-H
4106 110107 @"E-H*F+"C-H*F+"H-H
4107 47400 "O*L

; NO-ECHO

35 EDCOV=. -EMSG.
4110 106023 @"D-H*F+"A-H*F+"T-H
4111 101502 @"A-H*F+"[-H*F+"C-H
4112 116015 @"H-H*F+"A-H*F+"N-H
4113 27040 ". *L+"
4114 126023 @"L-H*F+"A-H*F+"T-H
4115 42400 "E*L

; DATA CHAN. LATE

4116 FBOVF=.
43 EBOVF=. -EMSG.
4116 103205 @"B-H*F+"U-H*F+"F-H
4117 112221 @"F-H*F+"E-H*F+"R-H
4120 164725 @"[-H*F+"O-H*F+"V-H
4121 111045 @"E-H*F+"R-H*F+"F-H
4122 120553 @"I-H*F+"L-H*F+"L-H
4123 0 0

; BUFFER OVERFILL

; INFORMATION MESSAGES

4124 MTITL=
4124 175675 @"_-H*F+ "^-H*F+ "^-H
4125 20052 " *L+ "*" *
4126 25052 "***L+ "*" *
4127 165504 @"[-H*F+ "[-H*F+ "E-H
4130 107110 @"D-H*F+ "S-H*F+ "I-H
4131 164610 @"[-H*F+ "M-H*F+ "I-H
4132 114363 @"Q-H*F+ "H-H*F+ "T-H
4133 161614 @"Y-H*F+ "]"-H*F+ "M-H
4134 151372 @"U-H*F+ "X-H*F+ "[-H
4135 106400 @"D-H*F+ "I-H*F+ "A-H
4136 114656 @"Q-H*F+ "N-H*F+ "O-H
4137 145150 @"S-H*F+ "T-H*F+ "I-H
4140 105532 @"C-H*F+ "[-H*F+ "[-H
4141 25052 "***L+ "*" *
4142 25015 "***L+15
4143 5000 12*L

; *** EDSI MIGHTY-MUX DIAGNOSTIC ***

4144 MSCORE=
4144 173156 @"^-H*F+ "T-H*F+ "O-H
4145 137526 @"P-H*F+ "[-H*F+ "W-H
4146 135043 @"O-H*F+ "R-H*F+ "D-H
4147 164705 @"[-H*F+ "O-H*F+ "F-H
4150 164116 @"[-H*F+ "C-H*F+ "O-H
4151 142232 @"R-H*F+ "E-H*F+ "[-H
4152 151104 @"U-H*F+ "S-H*F+ "E-H
4153 107537 @"D-H*F+ "[-H*F+ "`-H
4154 20000 " *L

; TOP WORD OF CORE USED =

4155 MSPED=
4155 175644 @"_-H*F+ "^-H*F+ "E-H
4156 112244 @"F-H*F+ "F-H*F+ "E-H
4157 105150 @"C-H*F+ "T-H*F+ "I-H
4160 152232 @"V-H*F+ "E-H*F+ "[-H
4161 130214 @"M-H*F+ "E-H*F+ "M-H
4162 135070 @"O-H*F+ "R-H*F+ "Y-H
4163 164130 @"[-H*F+ "C-H*F+ "Y-H
4164 104544 @"C-H*F+ "L-H*F+ "E-H
4165 165150 @"[-H*F+ "T-H*F+ "I-H
4166 130232 @"M-H*F+ "E-H*F+ "[-H
4167 177533 @"`-H*F+ "[-H*F+ "\-H

; EFFECTIVE MEMORY CYCLE TIME =

4170 MNSEC=
4170 163515 @"Z-H*F+ "[-H*F+ "N-H
4171 100656 @"A-H*F+ "N-H*F+ "O-H
4172 144202 @"S-H*F+ "E-H*F+ "C-H
4173 134643 @"O-H*F+ "N-H*F+ "D-H
4174 51400 "S*L

; 0 NANoseconds

4175 MBASE=
4175 175654 @"_-H*F+ "^-H*F+"M-H
4176 151372 @"U-H*F+"X-H*F+"[-H
4177 104715 @"C-H*F+"O-H*F+"N-H
4200 147056 @"T-H*F+"R-H*F+"O-H
4201 127500 @"L-H*F+"[-H*F+"A-H
4202 142200 @"R-H*F+"E-H*F+"A-H
4203 164040 @"[-H*F+"B-H*F+"A-H
4204 144232 @"S-H*F+"E-H*F+"[-H
4205 177533 @"\'-H*F+"[-H*F+"\'-H

; MUX CONTROL AREA BASE =

4206 MBLKL=
4206 175641 @"_-H*F+ "^-H*F+"B-H
4207 126702 @"L-H*F+"O-H*F+"C-H
4210 125513 @"K-H*F+"[-H*F+"L-H
4211 110646 @"E-H*F+"N-H*F+"G-H
4212 146372 @"T-H*F+"H-H*F+"[-H
4213 177533 @"\'-H*F+"[-H*F+"\'-H

; BLOCK LENGTH =

4214 MNPTS=
4214 175655 @"_-H*F+ "^-H*F+"N-H
4215 150601 @"U-H*F+"M-H*F+"B-H
4216 111072 @"E-H*F+"R-H*F+"[-H
4217 134272 @"O-H*F+"F-H*F+"[-H
4220 136721 @"P-H*F+"O-H*F+"R-H
4221 147132 @"T-H*F+"S-H*F+"[-H
4222 177533 @"\'-H*F+"[-H*F+"\'-H

; NUMBER OF PORTS =

4223 MECMD=
4223 175644 @"_-H*F+ "^-H*F+"E-H
4224 133144 @"N-H*F+"T-H*F+"E-H
4225 143502 @"R-H*F+"[-H*F+"C-H
4226 134614 @"O-H*F+"M-H*F+"M-H
4227 100643 @"A-H*F+"N-H*F+"D-H
4230 20050 " *L+"(
4231 147417 @"T-H*F+"Y-H*F+"P-H
4232 111507 @"E-H*F+"[-H*F+"H-H
4233 164256 @"[-H*F+"F-H*F+"O-H
4234 143507 @"R-H*F+"[-H*F+"H-H
4235 110557 @"E-H*F+"L-H*F+"P-H
4236 24400 ")*L

; ENTER COMMAND (TYPE H FOR HELP)

4237 MXMTR=
4237 175667 @"_-H*F+ "^-H*F+"X-H
4240 131161 @"M-H*F+"T-H*F+"R-H
4241 165772 @"[-H*F+"\'-H*F+"[-H
4242 0 0

; CR, LF, XMTR =

4243 MRCVR=
4243 26040 " , *L+"
4244 142125 @"R-H*F+"C-H*F+"V-H
4245 143537 @"R-H*F+"[-H*F+"\'-H
4246 20000 " *L

; , RCVR =

4247 MHELP=
4247 110557 @"E-H*F+"L-H*F+"P-H
4250 165310 @"[-H*F+"W-H*F+"I-H
4251 126572 @"L-H*F+"L-H*F+"[-H
4252 102232 @"B-H*F+"E-H*F+"[-H
4253 112724 @"F-H*F+"O-H*F+"U-H
4254 132172 @"N-H*F+"D-H*F+"[-H
4255 120672 @"I-H*F+"N-H*F+"[-H
4256 46525 "M*L+"U
4257 54104 "X*L+"D
4260 137513 @"P-H*F+"[-H*F+"L-H
4261 121123 @"I-H*F+"S-H*F+"T-H
4262 120646 @"I-H*F+"N-H*F+"Q-H
4263 164140 @"[-H*F+"D-H*F+"A-H
4264 146203 @"T-H*F+"E-H*F+"D-H
4265 20060 " *L+"O
4266 33055 "6*L+"-
4267 31067 "2*L+"7
4270 26467 "-*L+"7
4271 34415 "9*L+15
4272 172015 @"^-H*F+"A-H*F+"N-H
4273 107526 @"D-H*F+"[-H*F+"W-H
4274 142423 @"R-H*F+"I-H*F+"T-H
4275 111624 @"E-H*F+"]-H*F+"U-H
4276 137504 @"P-H*F+"[-H*F+"E-H
4277 107134 @"D-H*F+"S-H*F+"]-H
4300 30460 "1*L+"O
4301 31070 "2*L+"B
4302 27040 ". *L+"
4303 164540 @"[-H*F+"L-H*F+"A-H
4304 145172 @"S-H*F+"T-H*F+"[-H
4305 142222 @"R-H*F+"E-H*F+"S-H
4306 135063 @"D-H*F+"R-H*F+"T-H
4307 35040 " : *L+"
4310 104013 @"C-H*F+"A-H*F+"L-H
4311 46040 "L*L+"
4312 33461 "7*L+"1
4313 32055 "4*L+"-
4314 32465 "5*L+"5
4315 33055 "6*L+"-
4316 32062 "4*L+"2
4317 32062 "4*L+"2
4320 27000 ". *L

; HELP MESSAGE

; USE LISTING DATE HERE ****
; THEN ADJUST CHECKSUM BELOW

4321 171116 CKS7: 171116

; CHECKSUM

4322 2401 GO

4323 1400 .IND: ABORT ; OVERLAID BY MULTIPORT OUTPUT PGM.

4324 2401 GO

4325 1400 .INY: ABORT ; OVERLAID BY SYNC. PORT (EDS-302) TEST PGM.

; "Q" -- COMPOSITE EXERCISER FOR BOTH DISC AND MUX

; READS A BLOCK FROM DISC INTO CORE LOCATION X. THEN MAKES THE MUX
; TRANSFER IT FROM LOCATION X TO LOCATION Z. MEANWHILE IT READS THE SAME
; DISC BLOCK INTO CORE LOCATION Y. WHEN EITHER MUX OR DISC ARE DONE IT
; COMPARES THE BLOCKS FOR EQUALITY. MUX DRIVER IS INTERRUPT DRIVEN; DISC
; USES BZUP DRIVER AND IS NOT. USES MUX REAL TIME CLOCK TO MAKE SURE
; MUX IS ALIVE.

; *** REQUIRES BZUP (FOR DISC DRIVER) AT 5000 ***

```
4326 62677 IN. Q: IORST          ; START HERE
4327 4536          JSR          GCORC      ; CHECK PROGRAM INTEGRITY
4330 22523          LDA          0, @. BZUP ; CHECK THAT BZUP IS AT 5000
4331 24523          LDA          1, . BZUP+1
4332 30523          LDA          2, . BZUP+2
4333 106033         SLS          0, 1
4334 112433         SLE          0, 2
4335 452           JMP          NBZUP      ; NO BZUP AT 5000
4336 102400         SUB          0, 0      ; INITIALIZE CERTAIN PARAMETERS
4337 40522          STA          0, TIME
4340 42132          STA          0, @BASE
4341 20515          LDA          0, OKTIM
4342 24515          LDA          1, MXTIM
4343 122400         SUB          1, 0
4344 40127          STA          0, TSR+2
4345 20500          LDA          0, QX
4346 40500          STA          0, QY
4347 40500          STA          0, QZ
4350 6014          JSR          @. S960    ; SET MUX UP FOR 9600 BAUD
4351 60325          NIOP         MUX      ; AND START REAL-TIME CLOCK
4352 24472          LDA          1, QK    ; READ A BLOCK FROM DISC
4353 30472          LDA          2, QX
4354 6477          JSR          @. BZUP
4355 434           JMP          DRER      ; DISC READ ERROR
4356 4536          JSR          DOMUX     ; SET UP MUX FOR BLOCK TRANSFER
4357 20473          LDA          0, MASK  ; PREPARE TO ENABLE INTERRUPTS
4360 62077         MSKO         0
4361 4513          JSR          QISA     ; GET INTPT SERVICE ADDRESS
4362 54001         STA          3, 1
```

```
4363 20474 LOOP: LDA 0, MXTIM
4364 40125 STA 0, TSR
4365 30461 LDA 2, QY ; READ SAME BLOCK FROM DISC
4366 4425 JSR QADV ; ADVANCE DEST. THRU CORE
4367 50457 STA 2, QY
4370 24454 LDA 1, QK
4371 60177 INTEN
4372 6461 JSR @. BZUP
4373 416 JMP DRER ; DISC READ ERROR
4374 60277 INTDS
4375 30451 LDA 2, QY
4376 4477 JSR CMPAR ; COMPARE THE TWO BLOCKS
4377 20462 LDA 0, TIME ; MAKE SURE RTC IS WORKING
4400 24462 LDA 1, LASTM
4401 106414 SEQ 0, 1
4402 403 JMP . +3
4403 6012 JSR @. FALT
4404 3635 FNINT ; NO REAL-TIME CLOCK INTERRUPTS
4405 40455 STA 0, LASTM
4406 755 JMP LOOP

4407 6012 NBZUP: JSR @. FALT
4410 4674 FBZUP ; NO BZUP AT 5000 !?

4411 6012 DRER: JSR @. FALT
4412 4703 FDISC ; DISC READ ERROR

4413 20435 GADV: LDA 0, GINC ; ADVANCE DEST. BUFFER THRU CORE
4414 113000 ADD 0, 2
4415 20431 LDA 0, QY
4416 24431 LDA 1, QZ
4417 142414 SEQ 2, 0
4420 146415 SNE 2, 1
4421 772 JMP QADV
4422 20177 LDA 0, C400
4423 143000 ADD 2, 0
4424 24132 LDA 1, BASE
4425 106432 SGR 0, 1
4426 1400 JMP 0, 3
4427 24134 LDA 1, LPORT
4430 20133 LDA 0, BLKL
4431 107000 ADD 0, 1
4432 146032 SGE 2, 1
4433 760 JMP QADV
4434 20177 LDA 0, C400
4435 143000 ADD 2, 0
4436 24131 LDA 1, CORSZ
4437 146432 SGR 2, 1
4440 1400 JMP 0, 3
4441 30404 LDA 2, QX
4442 751 JMP QADV

4443 44121 GCKSM: 44121
```

; VARIABLES AND PARAMETERS NOT INCLUDED IN PROGRAM CHECKSUM

4444 4 GK: 4 ; DISC BLOCK TO BE READ
4445 11000 GX: 11000 ; REFERENCE LOCATION OF DISC BLOCK
4446 11400 QY: 11400 ; LOCATION WHERE DISC BLOCK IS REREAD
4447 12000 QZ: 12000 ; LOCATION WHERE MUX TRANSFERS BLOCK
4450 413 QINC: 413 ; AMOUNT BY WHICH DEST. IS INC'D EACH TIME

4451 50277 QPCON: 50277 ; PCON FOR 8-BIT, NO PARITY, 9600 BAUD
4452 175775 MASK: 175775 ; MASK OUT ALL EXCEPT MUX AND TTI
4453 5376 .BZUP: 5376 ; ENTRY TO BZUP'S READ ROUTINE
4454 600 600 ; MIN. LEGAL VALUE AT BZUP ENTRY POINT
4455 710 710 ; MAX. LEGAL VALUE AT BZUP ENTRY POINT

4456 13560 OKTIM: 13560 ; NUMBER OF R-T INTPTS. IN 1 MINUTE
4457 50 MXTIM: 50 ; MAX. TIME FOR A DISC TRANSFER
4460 110 110 ; MAX. TIME FOR A BLOCK TRANSFER THRU MUX
4461 0 TIME: 0 ; REAL-TIME IN 10 MSEC INTERVALS
4462 0 LASTM: 0 ; REAL-TIME SAVED HERE EACH TIME DISC IS DONE
4463 0 NCHA: 0 ; COUNTS NUMBER OF V'S TYPED ON ONE LINE
4464 72 LNPTH: 72 ; LINE LENGTH = 58 V'S, THEN CR, LF (1 HOUR)

; END OF VARIABLES

4465 54114 QCORC: STA 3, RTNAD ; CHECK PROGRAM INTEGRITY OF Q PROGRAM
4466 6203 JSR @.CORC
4467 4326 IN. Q
4470 4443 QCKSM
4471 4465 QCORC
4472 4726 QCKSN
4473 0 0

4474 473 QISA: JMP QINTS-1

4475 54114 CMPAR: STA 3, RTNAD ; COMPARE BLOCK AT (A2) WITH BLOCK X
4476 20747 LDA 0, QX
4477 40115 STA 0, PNTR
4500 20177 LDA 0, C400
4501 40116 STA 0, COUNT
4502 22115 CMPR1: LDA 0, @PNTR
4503 25000 LDA 1, 0, 2
4504 6011 TEST
4505 4716 FDATA ; DATA COMPARISON ERROR
4506 51000 STA 2, 0, 2 ; CHANGE THE WORD IN CORE
4507 151400 INC 2, 2
4510 10115 ISZ PNTR
4511 14116 DSZ COUNT
4512 770 JMP CMPR1
4513 2114 JMP @RTNAD


```

4514 54114 DOMUX: STA 3,RTNAD ; SET UP MUX FOR AUTO OUT + IN
4515 20127 LDA 0,TSR+2
4516 24740 LDA 1,OKTIM
4517 106032 SGE 0,1
4520 417 JMP DOMX2
4521 126400 SUB 1,1
4522 44127 STA 1,TSR+2
4523 10740 ISZ NCHA
4524 24737 LDA 1,NCHA
4525 30737 LDA 2,LENGTH
4526 132405 SUB 1,2,SNR ; END OF ONE LINE ?
4527 50734 STA 2,NCHA ; YES
4530 24543 LDA 1,C.V
4531 151235 MOVZR# 2,2,SNR ; NEXT TO LAST CHARACTER ?
4532 24047 LDA 1,C15 ; YES, MAKE IT A CARRIAGE RETURN
4533 151015 SNZ 2,2 ; LAST CHARACTER ?
4534 24046 LDA 1,C12 ; YES, CHANGE TO LINE FEED
4535 60225 NIOC MUX ; TURN OFF MUX, IN CASE OF T OPTION
4536 65111 DOAS 1,TT0
4537 30710 DOMX2: LDA 2,QZ
4540 4653 JSR QADV ; MOVE DESTINATION BUFFER
4541 50706 STA 2,QZ
4542 4421 JSR CWTB ; CONVERT TO BYTE ADDRESSES
4543 30137 LDA 2,RCVR ; SET UP RCVR CONTROL WORDS
4544 41004 STA 0,IBP.,2
4545 45006 STA 1,LIB.,2
4546 20065 LDA 0,C40KX ; ICW FOR AUTO INPUT
4547 41000 STA 0,ICW.,2
4550 30675 LDA 2,QX ; SET UP XMTR CONTROL XORDS
4551 4412 JSR CWTB
4552 30136 LDA 2,XMTR
4553 41005 STA 0,OBP.,2
4554 45007 STA 1,LOB.,2
4555 4253 JSR STOCW ; START MUX IN AUTO OUTPUT MODE
4556 20000 20000
4557 60325 NIOP MUX ; RESUME REAL-TIME CLOCK
4560 20700 LDA 0,MXTIM+1
4561 40126 STA 0,TSR+1
4562 2114 JMP @RTNAD

4563 140520 CWTB: NEGZL 2,0 ; CONVERT WORD ADDRESS TO BYTE ADDRESS
4564 100000 COM 0,0
4565 24505 LDA 1,C1000
4566 107000 ADD 0,1 ; A1 = LAST BYTE ADDRESS
4567 5400 JSR 0,3 ; (ALSO USED TO GET QINTS ADDRESS)

4570 62025 QINTS: DOB 0,MUX ; INTERRUPT SERVICE
4571 40120 STA 0,TS ; SAVE ACCUMULATORS AND CARRY
4572 44121 STA 1,TS+1
4573 50122 STA 2,TS+2
4574 54123 STA 3,TS+3
4575 101100 MOVL 0,0
4576 40124 STA 0,TS+4
4577 65477 INTA 1
4600 20045 LDA 0,C10
4601 106415 SNE 0,1 ; IS INTERRUPT FROM TTY INPUT ?
4602 2006 JMP @.WAIT ; YES

```

```
4603 20051 LDA 0, CMUX
4604 6011 TEST ; NO, IS IT FROM MUX ?
4605 3624 FINT ; ILLEGAL INTERRUPT OCCURRED
4606 60525 DIAS 0, MUX ; READ MUX STATUS WORD
4607 63025 DOC 0, MUX
4610 101113 SSN 0, 0 ; REAL-TIME INTERRUPT ?
4611 413 JMP CKIN ; NO
4612 10647 ISZ TIME
4613 14125 DSZ TSR
4614 403 JMP .+3
4615 6012 JSR @.FALT
4616 4711 FDTIM ; DISC TIMED OUT
4617 14126 DSZ TSR+1
4620 403 JMP .+3
4621 6012 JSR @.FALT
4622 3746 FTIMO ; MUX TIMED OUT
4623 10127 ISZ TSR+2
4624 24064 CKIN: LDA 1, C2OK
4625 107415 AND# 0, 1, SNR ; ANY INPUT DONE ?
4626 417 JMP CKOUT ; NO
4627 26137 LDA 1, @RCVR ; YES, CONFIRM RCVR ICW
4630 125112 SSP 1, 1
4631 403 JMP .+3
4632 6012 JSR @.FALT
4633 3526 FGARB ; GARBAGE COMING IN (SOME OTHER PORT)
4634 30137 LDA 2, RCVR
4635 21006 LDA 0, 6, 2
4636 25004 LDA 1, 4, 2
4637 6011 TEST
4640 3776 FMCWD ; IBP NOT EQUAL TO LIB
4641 30606 LDA 2, QZ
4642 4633 JSR CMPAR ; COMPARE BLOCKS
4643 4651 JSR DOMUX ; SET UP FOR NEXT TRANSFER
4644 416 JMP RSTOR

4645 24063 CKOUT: LDA 1, C1OK
4646 107415 AND# 0, 1, SNR ; ANY OUTPUT DONE ?
4647 413 JMP RSTOR ; NO
4650 30136 LDA 2, XMTR
4651 25001 LDA 1, OCW, 2
4652 125112 SSP 1, 1 ; WAS IT OUT PORT ?
4653 403 JMP .+3
4654 6012 JSR @.FALT ; NO, MUX STATUS WORD WRONG
4655 4004 FMSW
4656 21007 LDA 0, 7, 2
4657 25005 LDA 1, 5, 2
4660 6011 TEST ; IS OBP = LOB ?
4661 3776 FMCWD ; NO, FAULT
4662 20124 RSTOR: LDA 0, TS+4 ; RESTORE CARRY AND ACCUMULATORS
4663 101200 MOVR 0, 0
4664 20120 LDA 0, TS
4665 24121 LDA 1, TS+1
4666 30122 LDA 2, TS+2
4667 34123 LDA 3, TS+3
4670 60177 INTEN
4671 2000 JMP @0 ; RETURN TO INTERRUPTED PROGRAM
```

4672 1000 C1000:1000
4673 126 C.V: "V

; ADDITIONAL ERROR MESSAGES

4674 FBZUP=. ; NEED BZUP AT 5000
4674 132204 @"N-H*F+"E-H*F+"E-H
4675 107501 @"D-H*F+"[-H*F+"B-H
4676 55125 "Z*L+"U
4677 137500 @"P-H*F+"[-H*F+"A-H
4700 52040 "T*L+"
4701 32460 "5*L+"O
4702 163473 @"Z-H*F+"Z-H*F+"\-H

4703 FDISC=. ; DISC READ ERROR
4703 106422 @"D-H*F+"I-H*F+"S-H
4704 105521 @"C-H*F+"[-H*F+"R-H
4705 110003 @"E-H*F+"A-H*F+"D-H
4706 164221 @"[-H*F+"E-H*F+"R-H
4707 142721 @"R-H*F+"O-H*F+"R-H
4710 0 0

4711 FDTIM=. ; DISC TIMED OUT
4711 106422 @"D-H*F+"I-H*F+"S-H
4712 105523 @"C-H*F+"[-H*F+"T-H
4713 120604 @"I-H*F+"M-H*F+"E-H
4714 107516 @"D-H*F+"[-H*F+"O-H
4715 151173 @"U-H*F+"T-H*F+"\-H

4716 FDATA=. ; DATA COMPARISON ERROR
4716 106023 @"D-H*F+"A-H*F+"T-H
4717 101502 @"A-H*F+"[-H*F+"C-H
4720 134617 @"O-H*F+"M-H*F+"P-H
4721 101050 @"A-H*F+"R-H*F+"I-H
4722 144715 @"S-H*F+"O-H*F+"N-H
4723 164221 @"[-H*F+"E-H*F+"R-H
4724 142721 @"R-H*F+"O-H*F+"R-H
4725 0 0

4726 173440 QCKSN: 173440

; BIT MAP FOR STANDARD "SPECIAL CHARACTER" PROM

4740 PROM= 4740
11 .LOC PROM-
4740 .LOC PROM
2 .RDX 2

; .LOC OVERFLOW TEST

4740 177777 1111111111111111 ; \FIRST 32 CHARACTERS PRODUCE INTPT.
4741 177777 1111111111111111 ; \
4742 0 0000000000000000 ; \
4743 0 0000000000000000 ; \ (OUTPUT 0 - 177)
4744 0 0000000000000000 ; \
4745 0 0000000000000000 ; \
4746 0 0000000000000000 ; \
4747 170000 1111000000000000 ; /ASCII 174-177 = SPL. CHA.

4750 177777 1111111111111111 ; \
4751 177777 1111111111111111 ; \
4752 0 0000000000000000 ; \
4753 0 0000000000000000 ; \ (OUTPUT 200 - 377)
4754 0 0000000000000000 ; \
4755 0 0000000000000000 ; \
4756 0 0000000000000000 ; \
4757 170000 1111000000000000 ; /

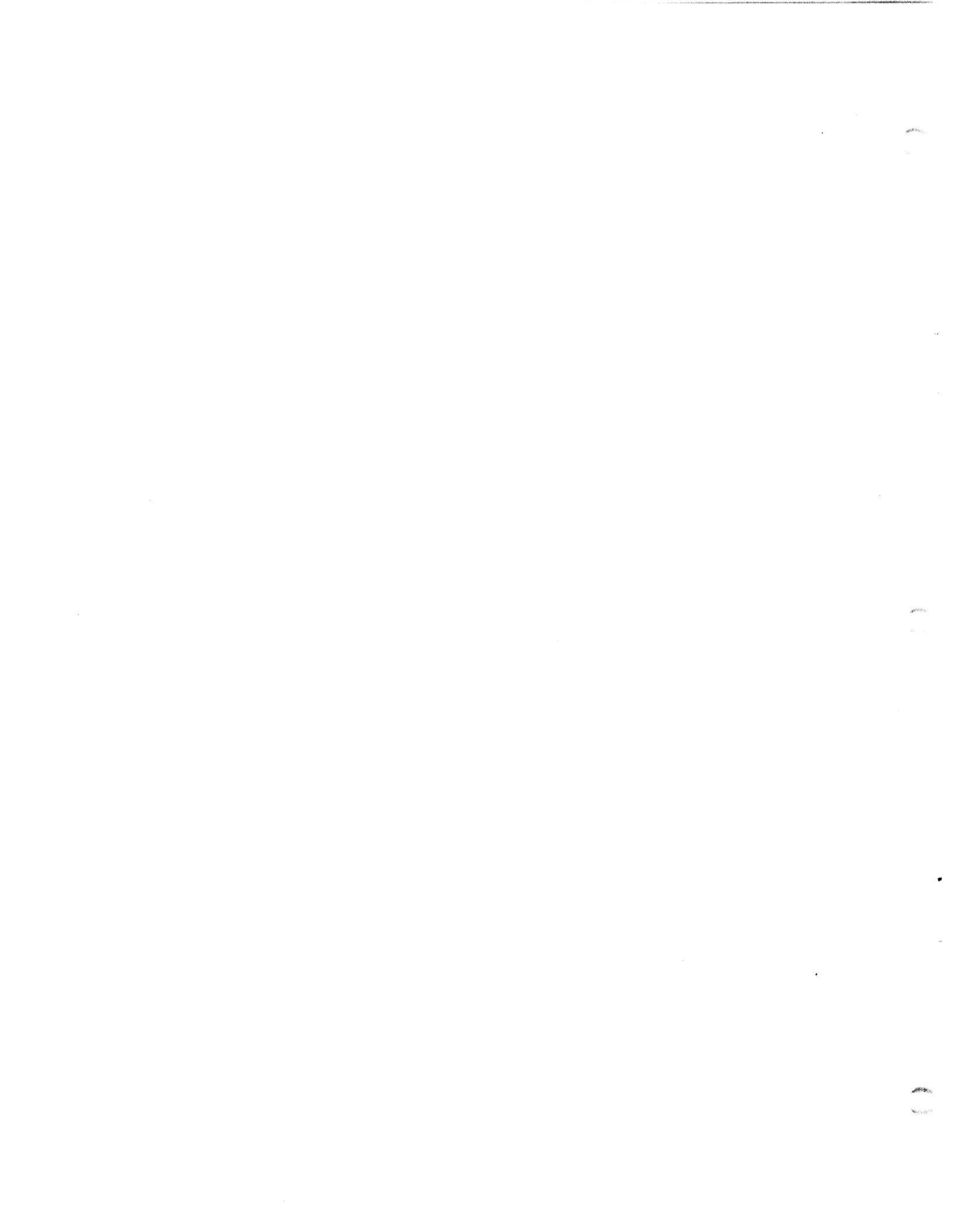
4760 0 0000000000000000 ; \7-BIT ASCII MODE SETS MSB = 1
4761 0 0000000000000000 ; \
4762 0 0000000000000000 ; \
4763 0 0000000000000000 ; \ (INPUT 0 - 177)
4764 0 0000000000000000 ; \
4765 0 0000000000000000 ; \
4766 0 0000000000000000 ; \
4767 0 0000000000000000 ; /

4770 177777 1111111111111111 ; \
4771 177777 1111111111111111 ; \
4772 0 0000000000000000 ; \
4773 0 0000000000000000 ; \ (INPUT 200 - 377)
4774 0 0000000000000000 ; \
4775 0 0000000000000000 ; \
4776 0 0000000000000000 ; \
4777 170000 1111000000000000 ; /

10 .RDX 8

.END ; MUXDP (MIGHTY MUX DIAGNOSTIC)

AO	110	A1	111	A2	112	A3	113	ABORT	1400
BASE	132	BFR	141	BG1	2372	BG2	2400	BLKL	133
BOTWD	100	BRESC	1546	BRNC2	1536	BRNCH	1530	C10	45
C1000	4672	C1002	2731	C101	56	C10K	63	C12	46
C12K4	1327	C140K	67	C15	47	C170K	70	C177	57
C2	156	C20	50	C20K	64	C3	40	C31	52



POINT 4 DATA CORPORATION

2569 McCabe Way / Irvine, California 92714 / (714) 863-1111