

200 000

PDP-12 LIBRARY

> a.

LINC-8 SIMULATOR TRAP PROCESSOR

Order DEC-12-SI1B-D from Program Library, Digital Equipment Corporation, Maynard, Massachusetts 01754 Price \$1.00

Direct comments concerning this document to:

Software Information Service **Digital Equipment Corporation** Maynard, Massachusetts 01754

CONTENTS

æ

		Page
1.	Abstract	1
2.	Equipment and Storage Requirements	1
2.1	Equipment	1
2.2	Storage	2
3.	Loading, Starting and Restarting	2
3.1	Loading Procedure	2
3.2	Normal Starting Procedure (Start 400)	2
3.3	Starting Procedure for Immediate GUIDE or LAP6–3L Loading and Starting (Start 20)	3
3.4	The Instruction Trap Enable Flip-Flop	3
3.5	Normal Restart (Start 400)	3
3.6	"User 20" Restart (Start 20)	4
3.7	GUIDE Load Restart (Start 700-717)	4
4.	Program Operation	4
4.1	Illegal Teletype Characters	5
4.2	Undefined Instruction Error Stop	5
5.	Internal Operation	5
5.1	Overview	5
5.2	Console Starts and Restarts	6
5.3	Operation of the Instruction Trap Hardware	6
5.4	Machine State Saving	7
5.5	Instruction Identification	7
5.6	TYP Processing	7
5.7	KBD Processing	7
5.8	Return To User Program	8
6.	LINC-8 Simulator Trap Processor Adaptions	8
6.1	Adaption To Process Additional LINC-8 OPR's	8
6.2	Example: Operating the XY12 Plotter Control with LINC or LINC-8 OPR's	9
6.3	Adaption for Convenient Trap Processor Loading from GUIDE or LAP6–3L Tapes	11
6.4	Adaption for Convenient Loading With a User Program From a DIAL Tape	12

CONTENTS (Cont)

J.

6.5	Adaption for Loading of Programs Other Than GUIDE and LAP6–3L By The Trap Processor	12
6.6	Adaption to Suppress Teletype Character Echoing	13
7.	Assembly Listing	

1. ABSTRACT

The LINC-8 Simulator Trap Processor handles Teletype input and output for LINC-8 and classic LINC programs when they are run on the PDP-12. It must be loaded into the PDP-12 core memory with any LINC-8 or classic LINC program which uses the keyboard, or any classic LINC program which uses the Teleprinter, in order for that program to run on the PDP-12.

The trap processor operates by using the PDP-12 Instruction Trap Facility to detect execution of either of the two LINC-8 Teletype input/output instructions by the user's program. It responds to user's execution of a Teletype instruction by executing coding to simulate the instruction's LINC-8 or classic LINC effect. After simulation of the instruction, the trap processor returns control to the user program.

Users may easily adapt the LINC-8 Simulator Trap Processor to their own purposes. Explicit instructions for a number of useful adaptions are provided in this document, along with enough information on the internal operation of the program to permit users to easily implement adaptions of their own invention. In this connection, attention is directed to the Dispatch Table Trap Processor, DEC-12-SI2A. It is a more suitable starting point for building extended trap processors which process a large number of different trapped instructions than is the LINC-8 Simulator Trap Processor.

An important limitation of the trap processor is that it is not interruptible. It may not be operated when the PDP-12 Program Interrupt is enabled.

This document applies to the machine readable program version bearing software product code DEC-12-SIIB-UA.

2. EQUIPMENT AND STORAGE REQUIREMENTS

2.1 Equipment

The LINC-8 Simulator Trap Processor is at present distributed in LINCtape form only, and therefore requires a PDP-12 with LINCtape control and at least one TU55 DECtape/LINCtape transport for program loading. The program itself will run on a minimum PDP-12.

The program operates correctly both in 4K PDP-12's and in PDP-12's having any amount of extended memory up to the 32K maximum total. Instruction traps originating in extended memory will be processed no differently than the same instruction traps originating in basic memory.

2.2 Storage

The program occupies most of the locations below 462, plus locations 700 to 717, all in PDP-12 memory segment 0.

3. LOADING, STARTING AND RESTARTING

3.1 Loading Procedure

The program is distributed on a DIAL tape. Load and start that copy of DIAL, and type into DIAL,

LO L8SIM.0)

(The symbol ↓ means Teletype LINE FEED. → means CARRIAGE RETURN.)

The LINC-8 Simulator Trap Processor will be loaded from the DIAL tape into the PDP-12 core memory, and the computer will halt.

3.2 Normal Starting Procedure (Start 400)

Press I/O Preset, and then Start 400. The program will turn on the Instruction Trap Enable Flip-Flop and halt with the Instruction Field set to 2 and the Data Field set to 3. Verify that the Instruction Trap Enable Flip-Flop is on by observing the console TRAP indicator. This indicator should be lit. If it is not, some kind of error has occurred. The error may be either a machine error or an operator error. Reload the trap processor and try again.

Now read in the user program. If the program is located on some specific block(s) of a LINCtape, mount the tape on either transport and execute an appropriate tape instruction from the console as if the machine were a LINC or a LINC-8. If the user program is a named file on a LAP6-3L or GUIDE tape, mount the tape on unit 0, set the LOCAL-OFF-REMOTE switch to REMOTE and press CONT. GUIDE or LAP6-3L* will be loaded, and the user program may be recalled using the usual GUIDE or LAP6 program loading procedure.

If the user program is on paper tape, read it in and start it using the usual paper tape loading and starting procedures, as described in the Binary Loader operating instructions, DEC-08-LBAA-D.

*LINC-8 and LINC users will recall that the GUIDE program starting procedure may be used with either GUIDE or LAP6-3L.

Switch the processor mode to the PDP-8 mode by executing the PDP instruction (octal:0002) before using the Binary Loader. Mode changing through use of I/O Preset in conjunction with the console Mode key should be avoided because I/O Preset clears the Instruction Trap Enable Flip-Flop.

3.3 Starting Procedure for Immediate GUIDE or LAP6-3L Loading and Starting (Start 20)

To automatically load and start a LINC-8 GUIDE or LAP6-3L tape along with the trap processor, load the trap processor from the DIAL tape as directed above, and then press I/O Preset, Start 20, rather than I/O Preset, Start 400. A GUIDE or LAP6-3L system will be read in from unit 0 and started.

This procedure duplicates the "Start 400" procedure given above, with the exception that the computer does not halt between the trap processor initialization and the loading and starting of the GUIDE or LAP6-3L system.

3.4 The Instruction Trap Enable Flip-Flop

Once a user program has been read in and started, the machine behaves like a LINC-8 or classic LINC with respect to Teletype input and output thereafter, but only if the Instruction Trap Enable Flip-Flop has been set. The PDP-12 Instruction Trap Enable Flip-Flop must be set in order for the trap processor to work. The state of this flip-flop is indicated by the TRAP light on the computer console. If the trap processor is loaded exactly as directed above, and if the user program is operated exactly as its instructions direct, the Trap Enable Flip-Flop will never be cleared, and will cause no problems. However, in practice it is sometimes cleared (by the operator pressing I/O PRESET for example), so some convenient methods for resetting it are included in the trap processor program in the form of the following restart procedures.

3.5 Normal Restart (Start 400)

If the Instruction Trap Enable Flip-Flop has been cleared, it may be set again (providing the trap processor has been loaded into core as directed above) by starting at location 400 in memory segment 0 (absolute address 00400). Note that the START 400 key may not be used for this unless the Instruction Field (IF) is set to 0 because START 400 takes the high order 5 bits of the starting address from the IF. Set 0400 into the Left Switches and use START LS, rather than Start 400. Use of this entry point sets the Trap Enable Flip-Flop and halts the computer. (Setting of the Trap Enable Flip-Flop may be confirmed by observing the console TRAP indicator.) Pressing continue after the computer has halted causes a transfer to location 400 in memory segment 2 (absolute address 04400), with the Data Field set to 3.

3.6 "User 20" Restart (Start 20)

Starting at location 20 in memory segment 0 (00020) sets the Trap Enable Flip-Flop and immediately transfers control to location 20 in memory bank 2 (absolute address 04020), with the Data Field set to 3. Note that the START 20 key may not be used for this unless the Instruction Field (IF) is set to 0 because START 20 takes the high order 5 bits of the 15 bit starting address from the IF. Set 0020 in the Left Switches and use Start LS, rather than using START 20.

3.7 GUIDE Load Restart (Start 700-717)

A third alternative is to start at any location between 700 and 717 in field 0. Use of any of these entry points sets the Trap Enable Flip-Flop and then loads and starts the LINC GUIDE or LAP6 system (if an appropriate tape is mounted on transport 0 and the LOCAL-OFF-REMOTE switch is set to REMOTE).

4. PROGRAM OPERATION

Once the LINC-8 or classic LINC user program and the LINC-8 Simulator Trap Processor have both been loaded into PDP-12 memory, the operating instructions for the user program apply, and the user program will behave as it would on a LINC-8 or classic LINC, and no special account need be taken of the fact that a PDP-12 rather than a LINC-8 or classic LINC is being used. There are a couple of minor exceptions to this. The PDP-12 console operates slightly differently from the LINC-8 console and the classic LINC console. Also, the characters which in the LINC are obtained by striking the CASE key and then some other key such as .,= and are obtained in the PDP-12 by striking a single Teletype key. The Teletype keys which are used to obtain the various LINC codes are indicated in the following diagram of the Teletype keyboard.



Figure 1. Location of LINC keys on the Teletype keyboard

An additional exception to strict classic LINC compatibility concerns "echoing" of keyboard characters. Characters typed into a user program running with the trap processor are automatically printed on the Teleprinter by the trap processor. Instructions for suppressing this feature are given in Section 6.6, Adaption to Suppress Teletype Character Echoing.

4.1 Illegal Teletype Characters

Some teletype keys, such as the semi-colon key, are not used for any LINC characters, either standard or special. These keys should not be struck when the trap processor is being used to run LINC or LINC-8 programs. If one of these illegal keys is struck, it is printed on the Teletype preceded by an up arrow, "[†]". Control is not returned to the user's LINC-8 or classic LINC program until a legal character is struck.

The following Teletype keyboard characters are illegal:

4.2 Undefined Instruction Error Stop

The trap processor halts at location 130 in memory segment 0 when any instruction other than a Teletype input or output instruction is given. The trapped instruction is contained in the accumulator, and the address of the instruction is in location 134 of memory segment 0.

5. INTERNAL OPERATION

5.1 Overview

Teletype input-output in LINC and LINC-8 programs involves only three instructions: Keyboard (mnemonic KBD; instruction code 515), Type Out (mnemonic TYP; instruction code 514) and Key Struck (mnemonic KST; instruction code 415). These are the only LINC-8 instructions whose operation involves the LINC-8 Simulator Trap Processor. The processing performed by the LINC-8 Simulator Trap Processor for the KBD and TYP instructions consists of detection of their execution by the user program, and execution of programming to reproduce in detail their LINC-8 or classic LINC effect. The KST instruction is a skip instruction which skips when the Teletype keyboard flag is set. This flag is set by the hardware when the operator strikes a Teletype key. The trap processor clears it when there is no keyboard character available for the user program to read in.

Processing for the trapped instructions KBD and TYP consists of three steps. Processing is initiated by the occurrence of an instruction trap. The first step consists of the identification of the instruction causing the trap. The second step is the execution of programming to simulate the LINC-8 effect of the particular trapped instruction. The final step is the return of control to the user program.

In addition to the "trap processor proper", which performs as described above, the program as distributed contains small amounts of code to implement the various console restarts described above.

5.2 Console Starts and Restarts

The program is initially loaded into memory segment 2, and relocates itself into segment 0 as soon as it is started. This technique is used because it leads to a particularly simple procedure for loading the program directly from LINCtape using the console functions. See Section 6.3, Adaption for Convenient Loading of the Trap Processor from GUIDE and LAP6-3L Tapes, for a description of this procedure. If the routine had to be loaded into segment 0 directly, the console procedure for loading it would be more complicated than it is because the operator would have to set one of the 5-bit Field Registers to 0 before execution of the tape read instruction in order to load data into segment 0. As is, the user may initially load the program into any memory segment for which memory is physically present.

Both the "Start 20" and "Start 400" program starting procedures execute the trap processor relocator routine at symbolic location SETUP. This routine relocates the trap processor from the current instruction field into memory segment 0. Notice that the relocator routine is not itself relocated, since it is used only once. After the trap processor has been relocated, the locations 700 through 717 are filled with "JMP GUIDE". This operation provides for restarting of the GUIDE system using the Start Left Switches function when any tape instruction is set into the Left Switches.

5.3 Operation of the Instruction Trap Hardware

The trap processor operates with the Instruction Trap facility enabled, so execution of any trappable instruction (except tape instructions) by the user's program causes a transfer of control to location 141 in memory segment 0 (absolute address 00141). Also, the low order 12 bits of the address of the instruction following the trap instruction is stored in location 140, and the contents of the Instruction Field and Data Field when the trap occurred is stored in the Save Field Register.

5.4 Machine State Saving

The instructions between 140 and symbolic location FETCH saves registers and indicators whose contents will be affected by the operation of the trap processor. They will be restored before control is returned to the user program. Notice that the Overflow flip-flop and memory location 0 are both saved. They are both affected by the operations in the trap processor.

5.5 Instruction Identification

The code from symbolic location FETCH to OP14 obtains the instruction causing the trap and transfers control to the subroutine which performs the LINC-8 function of the particular instruction.

5.6 TYP Processing

TYP and TYP I (OPR14 and OPR I 14) are processed identically, by the subroutine beginning at OPI14. Note that exit from the subroutine does not occur until the Teleprinter has completely finished printing the character. This assures that the printer will be ready to accept a new character the next time the subroutine is entered.

5.7 KBD Processing

The routine begins at tag OP15 and ends just before tag INIT. It has three functions. The functions are interrelated and are not performed in any simple sequence by the routine. Consequently, the following discussions of them do not reference specific pieces of code within the routine.

The first function performed is the translation of ASCII character code characters into LINC character code characters. (Part of the definition of the LINC KBD instruction is that characters which are read in by the instruction have the LINC character code.) This translation sometimes requires that two characters be passed to the user program when only one Teletype character has been typed. Some LINC "special" characters (?, =, u,,,, , [, _, :) did not have their own keys, and existed only by virtue of the convention that the "CASE" character followed by some other character was to be interpreted as a "special" character. The Teletype keyboard has a "SHIFT" facility whereby a single key may generate more than one character code, and, because of its ease of use, this "SHIFT" technique rather than the "CASE key" technique is used in the PDP-12 (and also the LINC-8) to generate the "special" characters. But since LINC programs "expect" to have special characters entered as two characters, and in general have no provision for accepting them in any other way, the trap processor must generate the appropriate pair of characters in response to the use of a single Teletype key.

The second function of this routine is the interpretation of the I bit. This bit, when raised, means, "Don't execute the next instruction until a character from the keyboard has been acquired. If no character has been typed, pause until the user types one."

The final function of the routine is the clearing of the keyboard flag. This is the flag which is sensed by the LINC KST instruction. Note that when two LINC characters are generated in response to one Teletype character, the flag is not cleared until after the second LINC character has been generated. This assures that the user program will "think" that the operator has struck the second character, and will execute the KBD instruction which collects that second character.

5.8 Return to User Program

The routine to return control to the user program begins at symbolic location RET and ends at OVN. The routine is entirely straightforward. It simply restores the various registers in the machine to their values prior to entry to the trap processor. Note that when the KBD instruction has been given the AC will get filled with some character code rather than with its contents at the time of trap processor entry.

6. LINC-8 SIMULATOR TRAP PROCESSOR ADAPTIONS

6.1 Adaption to Process Additional LINC-8 OPR's

LINC-8 and classic LINC installations often use instructions of the LINC OPR group for purposes other than control of the Teletype. For example, an installation may use OPR's 5, 6, 7, 10, 11 and 12 for controlling pen motion on an incremental plotter. Each OPR causes the pen to move one increment in one of six directions. In this section, we indicate how the LINC-8 Simulator Trap Processor may be "custom tailored" to process additional OPR's or other trapped instructions. In general, such modifications involve three steps:

a. extension of the trapped instruction identification routine to recognize trapped instructions other than the Teletype OPR's

- b. addition of a routine to simulate the effect of the LINC-8 or LINC OPR instruction; and
- c. return of control to the user program.

The extension to the trap identification routine consists simply of adding instructions to transfer control to a subroutine when one of the new OPR's is recognized. The new instructions should be added at

symbolic location OTHERS, and care should be taken to preserve the error stop which occurs when the trapped instruction is not recognized.

A routine which is entered when the new OPR's are given must be added to the program. The details of this routine are entirely dependent on the character of the new OPR.

Finally, the user program must be re-entered. This is accomplished by transferring control to symbolic location RET. Existing coding attends to the details of restarting the user program.

6.2 Example: Operating the XY12 Plotter Control with LINC or LINC-8 OPR's

Suppose that a LINC-8 or LINC user is operating an incremental plotter with 6 OPR instructions which work as follows:

OPR	5 (0505)	- lower pen
OPR	6 (0506)	– raise pen
OPR	7 (0507)	– move drum up 1 unit
OPR	10 (0510)	- move drum down 1 unit
OPR	11 (0511)	– move pen right 1 unit
OPR	12 (0512)	- move pen left 1 unit

Suppose further that each OPR "pauses" until the completion of the plotter operation caused by its execution. That is, execution of an OPR is not completed, and execution of the following instruction does not begin until the plotter operation caused by the OPR is completed. This "paused" mode of operation obtains whether or not the instruction's I bit is set.

The XY12 plotter control used the following PDP-8 mode commands:

PLSF	(6501)	– Skip on Plotter Flag
PLCF	(6502)	– Clear Plotter Flag
PLPU	(6504)	- Pen Up
PLPR	(6511)	– Pen Right
PLDU	(6512)	- Drum Up
PLDD	(6514)	- Drum Down
PLPL	(6521)	– Pen Left
PLPD	(6524)	– Pen Down

The plotter flag is set by the completion of a plotter command affecting the pen or the drum. It is cleared by the PLCF command only. The pen and drum commands do not affect the plotter flag.

The following modifications to the LINC-8 Simulator Trap Processor will enable it to run LINC or LINC-8 programs which operate the plotter in the manner described above:

1. Replace the instruction at symbolic location OTHER with the following instruction:

OTHER, JMP PLOT

2. Add this subroutine to the program just after symbolic location ENDX. (Actually, the routine could be inserted anywhere before TABE, which is the last location loaded into memory segment 0 by the loading sequence.)

ź.,

<pre>/SUBR TO RUN XY12 CONTROL FROM LINC /MODE USING TRAPPED CPRS. PLOT, STA I /SAVE INSTRUCTION PLINST, 0 BCL I /IGNORE INSTRUCTION I BIT I ADA I -OPR-12 APO I /OPR 12 OR LESS? JMP PLERR /NO. ERROR ADA I /YES. OK SO FAR 6 APO /OPR 5 OR MORE? JMP PLERR /NO. ERROR ADA I /YES. OBTAIN COMMAND ADD PLTAB-1 STC +1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLER, LDA PLINST</pre>	/		
<pre>/MODE USING TRAPPED OPRS. PLOT, STA I /SAVE INSTRUCTION PLINST, 0 BCL I /IGNOKE INSTRUCTION I BIT I ADA I -OPR-12 APO I /OPR 12 OR LESS? JMP PLERR /NO. ERROR ADA I /YES. OK SO FAR 6 APO /OPR 5 OR MORE? JMP PLERR /NO. ERROR ADA I /YES. OBTAIN COMMAND ADD PLTAB-1 STC +1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST</pre>	/SUBR TO RUN	XY12 CONTROL FROM 1	LINC
PLOT, STA I /SAVE INSTRUCTION PLINST, 0 BCL I /IGNORE INSTRUCTION I BIT I ADDA I -OPR-12 APO I /OPR 12 OR LESS? JMP PLERR /NO. ERROR ADA I /YES. OK SO FAK 6 APO /OPR 5 OR MORE? JMP PLERR /NO. ERROR ADA I /YES. OBTAIN COMMAND ADD PLTAB-1 STC +1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST	/MODE USING 7	RAPPED CPRS.	
PLINST, 0 BCL I I ADA I -OPR-12 APO I ADA I APO ADA I ADO PLERR ADA I ADD PLERR ADA I ADD PLERR ADA I ADD PLTAB-1 STC +1 HLT PLEF PLGO, HLT PLF JMP -1 LINC LMODE JMP RET / / / / / / / / / / / / /	PLOT,	STA I	/SAVE INSTRUCTION
BCL I /IGNORE INSTRUCTION I BIT I ADA I -OPR-12 APO I /OPR 12 OR LESS? JMP PLERR /NO. ERROR ADA I /YES. OK SO FAK 6 APO /OPR 5 OR MORE? JMP PLERR /NO. ERROR ADA I /YES. OBTAIN COMMAND ADD PLTAB-1 STC .+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST	PLINST,	Ø	
I ADA I -OPR-12 APO I JMP PLERR /NO. ERROR ADA I /VES. OK SO FAK 6 APO /OPR 5 OR MORE? JMP PLERR /NO. ERROR ADA I ADA I /VES. OBTAIN COMMAND ADD PLTAB-1 STC .+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		BCL I	/IGNORE INSTRUCTION I BIT
ADA I -OPR-12 APO I JMP PLERR ADA I ADA I ADA I ADA I ADA I ADA I APO JMP PLERR APO ADA I ADA I ADA I ADA I ADA I ADA I ADA I ADA I ADA PLTAB-1 STC +11 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP -1 LINC LMODE JMP RET /RETURN TO USER / / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		I	
-OPR-12 APO I /OPR 12 OR LESS? JMP PLERR /NO · ERROR ADA I /YES · OK SO FAK 6 APO /OPR 5 OR MORE? JMP PLERR /NO · ERROR ADA I /YES · OBTAIN COMMAND ADD PLTAB-1 STC ·+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP ·-1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		ADA I	
APO I /OPR 12 OR LESS? JMP PLERR /NO. ERROR ADA I /YES. OK SO FAK 6 APO /OPR 5 OR MORE? JMP PLERR /NO. ERROR ADA I /YES. OBTAIN COMMAND ADD PLTAB-1 STC .+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		-0PR-12	
<pre>JMP PLERR /N0. ERROR ADA I /YES. OK SO FAk 6 APO /OPR 5 OR MORE? JMP PLERR /N0. ERROR ADA I /YES. OBTAIN COMMAND ADD PLTAB-1 STC .+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST</pre>		APO I	/OPR 12 OR LESS?
ADA I /YES. OK SO FAR 6 APO /OPR S OR MORE? JMP PLERR /NO. ERROR ADA I /YES. OBTAIN COMMAND ADD PLTAB-1 STC .+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		JMP PLERR	/NO• ERROR
6 APO /OPR 5 OR MORE? JMP PLERR /NO · ERROR ADA I /YES · OBTAIN COMMAND ADD PLTAB-1 STC ·+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGF, JMP ·-1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		ADA I	/YES. OK SO FAK
<pre>APO /OPR 5 OR MORE? JMP PLERR /NO. ERROR ADA I /YES. OBTAIN COMMAND ADD PLTAB-1 STC .+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST</pre>		6	
<pre>/NO• ERROR ADA I /YES• OBTAIN COMMAND ADD PLTAB-1 STC •+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP •-1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST</pre>		APO	/OPR 5 OR MORE?
ADA I /YES.OBTAIN COMMAND ADD PLTAB-1 STC .+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLCF PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		JMP PLERR	/NO• ERROR
ADD PLTAB-1 STC •+1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP •-1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		ADA I	/YES. OBTAIN COMMAND
<pre>STC +1 HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLCF PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST</pre>		ADD PLTAB-1	
<pre>HLT /BECOMES ADD PLTAB + N STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST</pre>		STC •+1	
STC PLGO PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		НГТ	/BECOMES ADD PLTAB + N
PDP PMODE PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		STC PLGO	
PHGO, PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		PDP	
PLGO, PLCF PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		PMODE	
PLGO, HLT /PLPU,PLPR, ETC PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		PLCF	
PLSF JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST	PLG0,	НГТ	/PLPU,PLPR, ETC
JMP1 LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		PLSF	
LINC LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		JMP1	
LMODE JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		LINC	
JMP RET /RETURN TO USER / /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		LMODE	
/ /INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST		JMP RET	/RETURN TO USER
/INSTRUCTION NOT RECOGNIZED PLERR, LDA PLINST	1		
PLERR, LDA PLINST	/INSTRUCTION	NOT RECOGNIZED	
PLINST	PLERR,	LDA	
		PLINST	
JMP ERR /DO ERROR STOP		JMP ERR	/DO ERROR STOP
PDP		PDP	
PMODE		PMODE	
PLTAB, PLPD /PEN DOWN. OPR 5	PLTAB,	PLPD	/PEN DOWN. OPR 5
PLPU /PEN UP		PLPU	/PEN UP
PLDU /DRUM DOWN		PLDU	/DRUM DOWN
PLDD /DRUM UP		PLDD	/DRUM UP
PLPR /PEN KIGHT		PLPR	/PEN RIGHT
PLPL /PEN LEFT OPK 12		PLPL	/PEN LEFT .OPK 12
LINC		LINC	
LMODE		LMODE	

These modifications are most easily made by modifying the program's source and reassembling the program, as opposed to manually inserting the new instructions from the computer console. 6.3 Adaption for Convenient Loading of the Trap Processor from a GUIDE or LAP6-3L Tape

A particularly convenient way of using the trap processor with an existing GUIDE or LAP6-3L tape is to copy the program in exactly its present form onto some otherwise unused tape blocks and subsequently load it directly from these tape blocks using console functions. The advantage of this loading procedure is that it involves only the tape with the programs which are to be run with the trap processor. No separate DIAL tape is required for program loading purposes.

Procedures are given below for copying the trap processor onto a GUIDE or LAP6-3L tape and for subsequently reading it back into core for use with a LINC or LINC-8 program. Note that the program is stored in pairs of blocks which begin at blocks having block numbers which are multiples of 10 - e.g., blocks 10 and 11, or 230 and 231. This is done so that the tape group instructions WCG and RCG may be used to effect the tape operations involving the programs.

Notice also that this procedure may be used with LINCtapes other than GUIDE or LAP6-3L tapes.

Procedure to Copy the Trap Processor onto a GUIDE or LAP6-3L Tape:

- 1. Load the Trap Processor from a DIAL tape using the usual procedure.
- 2. Press I/O Preset.
- 3. Mount a GUIDE or LAP6-3L tape on unit 0.
- 4. Set in the Left and Right Switches:

Left Right 0705 1XX0

(XX0 is the first tape block of the pair of blocks in which the program is to be stored.)

5. Press DO. The Trap Processor will be written into the designated tape block.

Procedure to Load the Trap Processor from a GUIDE or LAP6-3L Tape:

- 1. Press STOP. Mount the GUIDE or LAP6-3L tape on unit 0.
- 2. Press I/O Preset.
- 3. Set in Left and Right Switches:

Left	Right
0701	1XX0

(XX0 is the first tape block of the pair of blocks onto which the program has been copied.)

4. Press DO. The Trap Processor will be read into memory segment 2.

5. Press Start 20. The Trap Processor will relocate itself from segment 2 into segment 0 and the GUIDE or LAP6-3L System will be loaded and started.

6.4 Adaption for Convenient Loading with a User Program from a DIAL Tape

The LINC-8 Simulator Trap Processor is designed to be loaded into memory ahead of user programs. In normal use, it is loaded into memory segment 2, and relocates itself into segment 0 as soon as it is started in order to leave segment 2 free for user programming. It is not difficult to modify the trap processor so that it may be loaded directly into segment 0. This modification may be made in the assembly source of the program, and this modified source may be assembled along with a DIAL source of the user program. Such a source may be obtained from a LAP6 source through use of the program CONVERT. The resulting program may then be filed and retrieved from LINCtape as a single binary program. However, once the program is loaded from LINCtape, one of the trap processor restarts must be executed in order to initialize the trap processor and set the Instruction Trap Enable Flip-Flop.

Production of a source modified for segment 0 loading consists of the following three steps:

1. Insert the pseudo-op SEGMNT 0 before the first line of code in the source program.

2. Replace the instructions in location 20 and symbolic location RST400 with "JMP INIT". This step prevents entry to the trap processor relocating program.

3. Delete all instructions after symbolic location TABE. These instructions comprise the trap processor relocation routine, which is no longer needed.

6.5 Adaption for Loading of Programs Other than GUIDE and LAP6-3L by the Trap Processor

It is easy to modify the trap processor to load programs other than GUIDE or LAP6-3L when using the automatic loading and starting feature. The code which implements the load and start feature begins at symbolic location GUIDE+1. It is quite straightforward:

GUIDE,	JMP INIT	/INITIALIZE TRAP PROCESSOR
	LDF 2	
	RDC	/READ GUIDE START BLOCK
	7400	/INTO QUARTER 3, SEGMENT 2
	LDF 3	
	LIF 2	
	DJR	
	JMP 1400	/START GUIDE

As an example, suppose that the trap processor is to be used with a LAP6 version other than LAP6-3L. Suppose the LINC-8 or classic LINC console procedure for loading and starting the other LAP6 version consists of executing the double-word tape instruction RCG 7300, and then pushing "START 20". The trap processor will perform this operation if the code at symbolic location GUIDE is replaced by the following:

GUIDE,	JMP LDF	INIT 2	/INITIALIZE TRAP PROCESSOR
	LDA RCG STA	Ĭ	/PUT RCG 7300 IN /4016 AND 4017
	2016 LDA 7300	Ι	
	STA 2017 LDF	3	
	L I F JMP	2 16	/JUMP TO 4016

The instruction RCG 7300 is inserted into locations 16 and 17 in memory bank 2 - i.e., 4016 and 4017 - and is immediately executed from these locations.* The next instruction is taken from location 4020. This is the starting location of the program.

6.6 Adaption to Suppress Teletype Character Echoing

The trap processor echoes characters typed by the computer operator on the console teleprinter in addition to transmitting them to the user program. This character "echoing" may be suppressed by removing seven instructions beginning at symbolic location L001. After modification, the section of the program near L001 should look like this:

L00;	STC	AC	STORE IT FOR TRANSFER
	IOB	UFC	VELEAR LEFT OVER DUFFER
	KRBA		/READ CHAR AND
			/CLEAR KEYBOARD FLAG
L001,	JMP	RET	/RETURN TO USER PROGRAM
/			
/ILLEGAL (CHARACTER		
RETX,	LDA I		/PRINT UP ARROW.
	•		
	•		
	•		
	•		

*One might ask here, "What happens if the RCG instruction fails at, say, block 301? The computer will try to execute the RCG again from the beginning, but the instruction will have been destroyed by the arrival in core of block 300." The answer is that when the PDP-12 tape control retries an instruction, it does not read the instruction from core a second time. It uses the same data it obtained when the instruction was read initially.

			*28			
0000			420 AL THC-8	STMILL AT	TOP TRAP	PROCESSOR, DEC-12-SI18,
0001			/LINC-U	STHOLA		
0002			(CORVET	CHT 1969	. DIGITA	L EQUIPMENT CORP.
0003			/MAYNAR	D. MASS		
0004			ZEALMAN	D1 11X331	•	
0005					7 MAY 19	69
0006			JU, LAN		V 1060	
0007			TREVISE		1040	
0010			TREATER	U S AUGI	121 1404	
0011			1005018	TON DEE	INT TIONS	FOR LINC MODE ASSEMBLY
0012			TUPERAT	IUN DEF	INT INST	PUCTONS
0013			JUE PUP	TO MULL	101 1831	NOCIONS.
0014			KCCASON	32		
0015			KRSABOD	34		
0016			KRUASOD	30		
0017			TSFAROD	91		
0920			TLSABOD	40		
0021			RMF A= 62	44		
0022			/			18
0023			SOME L	INC-0 U	EFINITION	15
8824			0PR=500	1		
0025			KBD=515)		
0026			-			
0027			● 2Ø		0.834	ADECOMES IND INIT
0030	0020	6463		JMP	STZI	ABECOMES JAP INTI
0031	0021	0643		LDF	3	
0032	0022	0602		LIF	2	
0033	8823	8086		DJK	0.0	ACO TO 26 IN SEGMENT 2
0034	0924	6020		JMP	20	/(64000)
9835						/ (0 4 0 2 0)
0936						
0037			/408 RE	START	0.0.0.0	ADDROUDE IND INIT
9049	8825	6466	RST400	JMP	ST400	ARECOMES THE INTE
0041	9926	9643		LDF	3	
0042	8827	8692		LIF	2	
8943	8838	0096		DJR		
0944	8831	0000		HLT		THE REAL THE OFCMENT O
8845	0032	6499		JMP	490	100 TO 400 IN SEGMENT 2
8845						/(84499)
8847			1			
0959			/START	GUIDE		
0051	8833	6356	GUIDE,	JMP	INIT	
9852	0034	0642		LDF	2	
0853	0035	0780		RDC		TREAD GUIDE START BLUCK
8854	0036	7400		7400		/INTO QUARTER SISEGMENT 2
8855	0037	\$643		LDF	3	
0056	9949	0602		LIF	2	
8857	8841	7488		JMP	1400	/START GUIDE
8868			1			
8861			/ERROR	STOP		
8862			+130			
9963	0130	0000	ERR,	HLT		/UNIDENTIFIED INSTRUCTION
9964	0131	6221		JMP	RET	/IN AC AT HLT. RETURNS TO
0065						/USER W AC CLEARED WHEN
9966						/CONTINUE PRESSED.
9947			/			
0007 0078			/MISC	CONSTAN	TS	
5 6 7 4			*134			
おお / 2	Ø134	8888	PC.	Ø		ADDRESS OF TRAPPED
10 10 / L						/INSTRUCTION
6697				7.0		
0973	9135	0070	M710 .	110		
0973 0974 0975	Ø135 Ø136	0070 0001	M70 P	10		

6 F

0140 0141 4242 STC AC /SAVE AC 0102 0142 0261 ROL I 1 /SAVE AC 0103 0143 4225 STC LC /GET H 0 11 0 REG 0104 0144 0005 0AC /GET H 0 11 0 REG 0105 0147 2136 ADD ONE /GET H 0 BIT 0110 0150 4232 STC 0 /SAVE IT 0111 0151 2136 ADD ONE /GET OVERFLOW BIT 0112 0152 0454 FLO /GET OVERFLOW BIT 0111 0151 0155 0807 SET 17 /SAVE IT SAVE LICATION # 0111 0157 0802 PDP /INSTRUCTION IDENTIFICATION. ////////////////////////////////////	0076 0077			/ /TRAP	ENTRY	ANO	MACHINE	STATE SAVING
0142 0142 0142 0142 0142 0142 0142 0142 0144	0100	0141	4242	*141	STC		٨C	ISAVE AC
0103 0143 4225 STC LC /GET HO 11 0 REG 0104 0144 0005 0AC /GET HO 11 0 REG 0106 0146 0446 0245 0AC ONE /GET LO BIT 0106 0450 0432 STC O /SAVE IT 0111 0150 0437 CLA OD /GET OVERFLOW BIT 0112 0155 0454 CLC /JAST OVE /SAVE IT 0113 0155 0454 CLC OVE /SAVE IT SAVE IT<	0101	0142	0261		ROL	T	1	/SAVE I INK
0104 0105 04C /GET H 0 11 0 REG 0105 0145 0241 RQL 1 /GET L 0 BIT 0107 0147 0150 4232 STC 0 /SAVE IT 0110 0150 4232 STC 0 /SAVE IT 0111 0151 2136 ADO ONE /GET OVERFLOW BIT 0113 0153 0421 STC 0 /SAVE IT.@=OFF.1 0113 0153 0401 CLP /SAVE IT.@=OFF.1 0115 0155 0657 SET 17 /SAVE IT.@=OFF.1 0116 0156 0207 P //INSTRUCTION IDENTIFICATION. 0117 ////////////////////////////////////	0103	0143	4225		STC	•	Ĩ.c	
0100 01440 02447 2136 ADD ONE 0100 0150 4232 STC 0 /SAVE IT 0111 0151 2136 ADD ONE /SAVE IT 0112 0152 4232 STC 0 /SAVE IT 0113 0153 0011 CLR /CET OVERFLOW BIT 0114 0154 4222 STC OVL /SAVE IT.Ø=OFF, 1 0115 0155 0800 Ø // // /SAVE IT.Ø=OFF, 1 0114 0154 0207 SET 17 /SAVE LOCATION Ø 0121 0157 0002 PDO // // 0122 0157 0002 PDO // // 0123 4168 6234 FETCH, RIB // // // 0124 4161 0335 ANO MFØ // // // 0124 4167 324 DCA // // // // // // // 0125 4167 14	0104	Ø144	0005		QAC		1	/GET H O 11 Q REG BITS
0107 0147 2136 ACD ONE 0110 0151 2136 ADD ONE 0111 0151 2136 ADD ONE 0112 0152 0454 FLO /GET OVERFLOW BIT 0113 0153 0011 CLR SET 17 /SAVE LT.0#OFF, 1 0114 0154 4222 STC OVL /SAVE LT.0#OFF, 1 0115 0155 0007 0 POP POP 0120 //INSTRUCTION IDENTIFICATION. POP 0121 0157 0202 POP POP 0122 4168 0234 FETCH, RIB /READ INTERRUPT BI 0124 4161 0335 ANO M78 /GET INSTRUCTION ID 0124 4162 371 TAD CDFX /MAKE IT INSTRUCTION ID 0124 4167 334 DCA PC /TREAD INTERRUPT BI 0133 4166 1740 CACT NAN M78 /CET INSTRUCTION ID 0135 4171 6201 DFX <t< td=""><td>Ø1Ø5 Ø1Ø6</td><td>Ø145 Ø146</td><td>0241</td><td></td><td>QLZ</td><td></td><td>1</td><td>/GET L O BIT</td></t<>	Ø1Ø5 Ø1Ø6	Ø145 Ø146	0241		QLZ		1	/GET L O BIT
8111 0151 2136 ADD ONE 8113 0153 0011 CLR /GET OVERFLOW BIT 8114 0154 4222 STC OVL /SAVE IT.8=OFF.1 8116 0156 0020 0 0 ////////////////////////////////////	Ø1Ø7 Ø11Ø	Ø147 Ø15Ø	2136 4232		ADD Stc		ONE Q	/SAVE IT
3113 8153 8011 CLR VI /SAVE IT.8=OFF, 1: 8114 8154 4222 STC OVL /SAVE IT.8=OFF, 1: 8115 8157 SET 17 /SAVE LOCATION # 8116 0156 8007 B ////////////////////////////////////	0111	Ø151 Ø152	2136 0454		ADD	ONE		ZGET OVERFLOW BIT IF ON
0114 0154 4222 SIC UVL /SAVE LICATION 0 0116 0156 0000 0	0113	0153	0011		CLR		.	
Ø115 Ø155 Ø157 SET 17 /SAVE LUCATION D Ø116 Ø156 ØØ97 SET 17 /SAVE LUCATION D Ø120 //INSTRUCTION IDENTIFICATION. POP Ø121 Ø157 ØØ92 PODE Ø122 4160 6234 FETCH, RIB /READ INTERRUPT BI Ø124 4161 Ø335 ANO M79 /CET INSTRUCTION ID Ø125 4162 1371 TAD CDFX /MAKE LI INTO A CI Ø126 4163 3364 DCA ,*1 /ØECOMES CDF N Ø138 4165 7240 STA /ØET AC TO -1 //SET AC TO -1 Ø131 4166 1140 TAD I PC //RAPPED INSTRUCTI //CAUSING TRAP Ø133 4171 6201 CDFX, CDF /SET AC TO -1 Ø133 4172 6141 LNOC //CAUSING TRAP Ø143 0173 1460 SAE I /IS IT KBO? Ø142 0173 1460 S	0114	0154	4222		SIC			SAVE 11.0=UFF. 1=UN
Disc Disc Disc Disc Disc Disc 0117 /instruction identification. /instruction identification. POP 0121 0157 0002 PMODE 0122 4160 0335 ANO MTA /CET INSTRUCTION I 0124 4161 0335 ANO MTA /CET INSTRUCTION I 0126 4163 3364 DCA .+1 0127 4164 7402 /BECOMES CDF N 0138 4166 1740 TAD CCA .+1 0137 4164 7402 7402 /BECOMES CDF N 0138 4166 1140 TAO 140 /COMPUTE ADDRESS 0133 4166 1734 DCA PC /TRAPPED INSTRUCTION 0136 4172 6141 LINC	0115	0100	0007		361		1/	SAVE LUCATION D
0120 /INSTRUCTION IDENTIFICATION. 0121 0157 0002 POP 0123 4160 6234 FETCH, RIB /READ INTERRUPT BI 0124 4161 0335 ANO M7Ø /GET INSTRUCTION J 0125 4162 1371 TAO COFX /MAKE IT INTO A CO 0126 4163 3364 DCA .*1 0130 4165 7240 STA /SET AC TO -1 0131 4166 1440 TAO 140 /COMPUTE ADDRESS CDF N 0133 4167 7334 DCA PC /TRAPPED INSTRUCTION 0133 4170 1734 TAO I PC /OBTAIN INSTRUCTION 0134 4167 3334 DCA PC /TRAPPED INSTRUCTION /CAUSING TRAP 0135 4171 6201 CDFX, CDF /SET ACTO -1 /SET ACTO -1 0136 4172 6141 LINC /CAUSING TRAP /SET ACTO -1 /SET ACTO -1 0136 4171 6201 CDFX, CDF /SET ACTO -1 /SET	0117	0100	0000	/	U			
0121 0157 0002 PDP 0122 PMODE 0123 4160 6234 FETCH, RIB /READ INTERRUPT BI 0124 4161 0335 ANO M70 /GET INSTRUCTION I 0125 4162 1371 TAO COFX /MAKE IT INTO A CI 0126 4163 3364 DCA .+1 /BECOMES CDF N 0130 4165 7240 STA /SET AC TO -1 0131 4166 1140 TAO 140 /COMPUTE ADDRESS (DF N 0132 4167 7334 DCA PC /TRAPED INSTRUCT 0133 4170 1734 TAO I PC /OBTAIN INSTRUCT 0136 4172 6141 LINC /CAUSING TRAP 0137 1460 SAE I /IS IT KBO? 0142 0175 0467 SKP /NO 0143 0176 6250 JMP OOKBD /YES, 0144 0177 1460 SAE I /OPR I 14? 0150 0203 1460 SAE I /OPR	0120				UCTION		ENTIFICA	TION.
0122 PMODE 0123 4160 6234 FETCH, RIB /READ INTERRUPT BI 0124 4161 0335 AND M70 /GET INSTRUCTION J 0125 4162 1371 TAD CDFX /MAKE IT INTO A CI 0126 4163 3364 DCA *1 //MAKE IT INTO A CI 0126 4163 3364 DCA *1 //MAKE IT INTO A CI 0130 4165 7240 STA /SET AC TO -1 0131 4166 1140 TAO I Y /COMPUTE ADDRESS CI 0132 4167 3334 DCA PC /TRAPPED INSTRUCTI 0133 4170 1734 TAO I PC /OBTAIN INSTRUCTI 0134 4171 6281 CDFX, CDF /SET OATA FIELD TO 0136 4171 6281 DKBD /YES //SET OATA FIELD TO 0136 4171 6281 DKBD /YES //SET OATA FIELD TO 0141 0173 1460 SAE I /IS IT KBD1? 0142 0173 1460 <	0121	0157	0002		POP			
0123 4160 0234 FETCH, RIB //READ INTERRUPT BI 0124 4161 0335 ANO M70 /GET INSTRUCTION INTO A CI 0125 4162 1371 TAD COFX /MAKE IT INTO A CI 0126 4164 7402 7402 /AKE IT INTO A CI 0130 4165 7240 STA /SET AC TO ~1 0131 4166 1140 TAO 140 /COMPUTE ADORESS CI 0132 4167 3334 DCA PC /TRAPPED INSTRUCTI 0133 4170 1734 TAO I PC /OBTAIN INSTRUCTI 0135 4171 6201 CDFX, CDF /SET OATA FIELD TO 0136 4172 6141 LINC /IS IT KBO? //IS IT KBO? 0144 0173 1460 SAE I /IS IT KBO? //IS IT KBO? 0144 0175 6467 SKP //NO. //IS IT KBO ? 0144 0175 6467 SKP /NO. //IS IT KBO ? 0145 0200 0535 KBD I //IS IT KBO ?<	0122				PMOD	E		
0124 4161 0335 ANO M70 /GET INSTRUCTION I 0125 4162 1371 TAD CDFX /MAKE IT INTO A CI 0126 4163 3364 DCA ,*1 /MECOMES CDF N 0130 4165 7240 STA /SET AC TO -1 0131 4166 1140 TAO 140 /COMPUTE ADORESS (0132 4167 3334 DCA PC /TRAPPED INSTRUCT 0133 4170 1734 TAO I PC /OBTAIN INSTRUCTION I 0135 4171 6201 CDFX, CDF /SET AC TO -1 0135 4171 6201 CDFX, CDF /SET AC TO -1 0136 4172 6141 LINC /CAUSING TRAP 0136 4172 6141 LINC /SET AC TO -1 0140 0173 1460 SAE I /IS IT KBO? 0141 0174 4515 KBD /NO 0144 0177 1460 SAE I /IS IT KBO I? 0144 0201 0467 <t< td=""><td>0123</td><td>4160</td><td>6234</td><td>FETCH,</td><td>RIB</td><td></td><td></td><td>/READ INTERRUPT BUFFER</td></t<>	0123	4160	6234	FETCH,	RIB			/READ INTERRUPT BUFFER
8125 4162 1371 TAD COFX /MAKE IT INTO A CI 8126 4163 3364 DCA .*1 9127 4164 7402 7402 /BECOMES CDF N 9130 4165 7240 STA /SET AC TO -1 9131 4166 1140 TAO I40 /COMPUTE ADORESS CDF N 9131 4166 1140 TAO I40 /COMPUTE ADORESS CDF N 9131 4166 1140 TAO I40 /COMPUTE ADORESS CDF N 9132 4167 3334 DCA PC /TRAPPED INSTRUCT 9133 4170 1734 TAO I PC /OBTAIN INSTRUCTION 9135 4171 6201 CDFX, CDF /SET OATA FIELD TO 9136 4172 6141 LINC //SET OATA FIELD TO 9136 4171 6201 CDFX, CDF /SET OATA FIELD TO 9140 9173 1460 SAE I /IS IT KBO? //SET 9144 9173 1460 SAE I /NO, //SET, <td>0124</td> <td>4161</td> <td>0335</td> <td></td> <td>ANO</td> <td></td> <td>M70</td> <td>/GET INSTRUCTION FIELD</td>	0124	4161	0335		ANO		M70	/GET INSTRUCTION FIELD
0126 4163 3364 DCA .+1 0137 4164 7402 7402 /BECOMES CDF N 0130 4165 7240 STA /SET AC TO -1 0131 4166 1140 TAO 140 /COMPUTE ADDRESS (0132 4167 3334 DCA PC /TRAPPED INSTRUCT 0133 4170 1734 TAO I <pc< td=""> /OBTAIN INSTRUCT 0134 171 6201 CDFX, CDF /SET OATA FIELD T(0135 4171 6201 CDFX, CDF /SET OATA FIELD T(0136 4172 6141 LINC /SET OATA FIELD T(0137 1460 SAE I /IS IT KBO? 0141 0174 4055 KBD /NO 0142 0175 467 SKP /NO 0143 0176 6250 JMP OKBD /YES, 0144 0177 1460 SAE I /IS IT KBO I? 0146 0203 1460 SAE I /OPR I 142? 0151 <td< td=""><td>0125</td><td>4162</td><td>1371</td><td></td><td>TAD</td><td></td><td>CDFX</td><td>/MAKE IT INTO A CDF N</td></td<></pc<>	0125	4162	1371		TAD		CDFX	/MAKE IT INTO A CDF N
0127 4164 7402 7402 7402 7402 7402 7402 751 AC TO -1 0131 4166 1140 TAO 140 751 AC TO -1 0131 4166 1140 TAO 140 751 AC TO -1 0132 4167 3334 DCA PC 778APPED INSTRUCTI 0133 4170 1734 TAD I PC 778APPED INSTRUCTI 0135 4171 6201 CDFX, CDF /SET AC TO -1 0136 4172 6141 LINC /CAUSING TRAP 0136 4172 6141 LINC /CAUSING TRAP 0140 0173 1460 SAE I /IS IT KB0? 0141 0174 0515 KBD /NO /NO 0142 0175 0467 SKP /NO /NO 0143 0176 6250 JMP OOKBO YES. 0144 0177 1460 SAE I /NO 0145 0202 6256 JMP OKBO YES.	Ø126	4163	3364		DCA		.+1	
0130 4105 7240 STA 7261	0127	4104	7402		7402			VBECOMES CUP N
0131 1100 1100 1100 700 <	0130	4100	1140		TA		140	COMPLETE ADORESS OF
0132 4107 0334 TAD I PC /IRATAN INSTRUCTIO 0134 //2 111 6201 CDFX, CDF /SET OATAFIED INSTRUCTIO 0135 4171 6201 CDFX, CDF /SET OATAFIED INSTRUCTIO 0136 4172 6141 LINC LNODE /INATAFIED /INATAFIED INSTRUCTIO 0140 0173 1460 SAE /INATAFIED /INAT	0131	4167	2324		DCA		140	TRAPPED INSTRUCTION
0134 ////////////////////////////////////	0133	4170	1734		TAD	T	PC	ZOBTAIN INSTRUCTION
0135 4171 6201 CDFX, CDF /SET 0ATA FIELD T(0136 4172 6141 LINC LMODE ////////////////////////////////////	0134	4170	1/04		TAD	1	10	ZCAUSING TRAP
Ø136 4172 6141 LINC Ø137 LMODE LMODE Ø140 Ø173 1460 SAE I /IS IT KBO? Ø140 Ø173 1460 SAE I /IS IT KBO? Ø141 Ø174 Ø515 KBD /NO Ø142 Ø175 Ø467 SKP /NO Ø143 Ø176 6250 JMP OOKBD /YES. Ø144 Ø177 1460 SAE I /IS IT KBO I? Ø145 Ø200 Ø535 KBD I /IS IT KBO I? Ø146 Ø201 Ø467 SKP /NO. /IS IT KBO I? Ø146 Ø202 6250 JMP DOKBO /YES. Ø150 Ø203 1460 SAE I /OPR I 14? Ø152 Ø206 6212 JMP OP14 /YES Ø154 Ø207 1460 SAE I /OPR 14? /OTFR 14? Ø155 Ø210 Ø514 OPR 14 /OFR 14? /OTFR 14? Ø164 Ø212 <td< td=""><td>0135</td><td>4171</td><td>6201</td><td>CDEX</td><td>COF</td><td></td><td></td><td>SET DATA FIELD TO Ø</td></td<>	0135	4171	6201	CDEX	COF			SET DATA FIELD TO Ø
0137 LMODE /IS IT KB07 0140 0173 1460 SAE I /IS IT KB07 0141 0174 0515 KBD /NO 0142 0175 0467 SKP /NO 0143 0176 6250 JMP 00KBD /YES. 0144 0177 1460 SAE I /IS IT KB0 I? 0145 0200 0535 KBD I /NO. 0146 0201 0467 SKP /NO. 0147 0202 6250 JMP D0KB0 /YES. 0150 0203 1460 SAE I /OPR I 147 0151 0204 0534 OPR I /OPR 147 0152 0205 0467 SKP /NO /OIT 0153 0206 6212 JMP OP14 /YES 0154 0207 1460 SAE I /OPR 147 0155 0210 0514 OPR 14 0156 0210 0514 OPR </td <td>0136</td> <td>4172</td> <td>6141</td> <td>00</td> <td>I INC</td> <td></td> <td></td> <td></td>	0136	4172	6141	00	I INC			
Ø14Ø Ø173 146Ø SAE I /IS IT KBO? Ø141 Ø174 Ø515 KBD /NO Ø142 Ø175 Ø467 SKP /NO Ø143 Ø176 6250 JMP OOKBD /YES. Ø144 Ø177 146Ø SAE I /IS IT KBO I? Ø144 Ø177 146Ø SAE I /IS IT KBO I? Ø146 Ø200 Ø535 KBD I //is it kBO i? Ø146 Ø201 Ø467 SKP /NO. Ø147 Ø202 6250 JMP DOKBO /YES. Ø150 Ø203 146Ø SAE I /OPR I 14? Ø151 Ø204 Ø534 OPR I I4 Ø153 Ø206 6212 JMP OP14 /YES Ø154 Ø207 146Ø SAE I /OPR 14? /OPR 14? Ø155 Ø210 Ø514 OPR I4 /OUTPUT, GET CHARA Ø160 Ø211 613Ø OP144, LOA /OUTPUT, GET CHARA	0137				LMOD	E		
Ø141 Ø174 Ø515 KBD /NO Ø142 Ø175 Ø467 SKP /NO Ø143 Ø176 6250 JMP OOKBD /YES. Ø144 Ø177 1460 SAE I /IS IT KBO I? Ø145 Ø200 Ø535 KBD I /NO. Ø146 Ø201 Ø467 SKP /NO. Ø146 Ø201 Ø467 SKP /NO. Ø146 Ø202 Ø203 J460 SAE I /OPR I 14? Ø150 Ø203 1460 SAE I /OPR I 14? Ø151 Ø204 Ø534 OPR I 14 /NO Ø153 Ø206 6212 JMP OP14 /YES Ø154 Ø207 1460 SAE I /OPR 14? /OFR 14? Ø156 Ø211 6130 OTHERS, JMP ERR /PUT JMP TO Ø165 Ø212 1000 OP14, LOA /OUTPUT, GET CHARA Ø160 Ø212 1000 OP14, LOA /OUTPUT, GET CHARA Ø163 Ø214 Ø500 <td>0140</td> <td>Ø173</td> <td>1460</td> <td></td> <td>SAE</td> <td>I</td> <td></td> <td>/IS IT KBO?</td>	0140	Ø173	1460		SAE	I		/IS IT KBO?
Ø142 Ø175 Ø467 SKP /N0 Ø143 Ø176 6250 JMP OOKBD /YES. Ø144 Ø177 1460 SAE I /IS IT KBO I? Ø145 Ø200 Ø535 KBD I /NO. Ø146 Ø201 Ø467 SKP /NO. Ø146 Ø202 6250 JMP DOKBO /YES. Ø150 Ø203 1460 SAE I /OPR I 14? Ø151 Ø204 Ø534 OPR I 14 Ø152 Ø205 Ø467 SKP /NO Ø153 Ø206 6212 JMP OP14 /YES Ø154 Ø207 1460 SAE I /OPR 14? Ø155 Ø210 Ø514 OPR 14 Ø156 Ø211 6130 OTHERS, JMP ERR /PUT JMP TO Ø161 Ø212 1000 OP14, LOA /OUTPUT, GET CHARA Ø162 Ø213 Ø242 AC /INSTRUCTIONS HERE Ø163 Ø214	Ø141	Ø174	Ø515		KBD			
Ø143 Ø176 6250 JMP OOKBD /YES. Ø144 Ø177 1460 SAE I /IS IT KBO I? Ø145 Ø200 Ø535 KBD I /IS IT KBO I? Ø146 Ø201 Ø467 SKP /NO. Ø147 Ø202 6250 JMP DOKBO /YES. Ø150 Ø203 1460 SAE I /OPR I 14? Ø151 Ø204 Ø534 OPR I 14 Ø152 Ø205 Ø467 SKP /NO Ø153 Ø206 6212 JMP OP14 /YES Ø154 Ø207 1460 SAE I /OPR 14? Ø156 Ø210 Ø514 OPR 14 /OUT JMP TO Ø156 Ø211 6130 OTHERS, JMP ERR /PUT JMP TO Ø160 Ø212 1000 OP14, LOA /OUTPUT, GET CHARA Ø162 Ø213 Ø242 AC /OUTPUT, GET CHARA Ø163 Ø214 Ø500 IOB /OUTPUT IT <tr< td=""><td>0142</td><td>Ø175</td><td>Ø467</td><td></td><td>SKP</td><td></td><td></td><td>/NO</td></tr<>	0142	Ø175	Ø467		SKP			/NO
0144 0177 1460 SAE I /IS IT KBO I 0145 0200 0535 KBO I /NO. /NO. 0146 0201 0467 SKP /NO. /NO. 0147 0202 6250 JMP DOKBO /YES. 0150 0203 1460 SAE I /OPR I 147 0151 0204 0534 OPR I 14 0152 0205 0467 SKP /NO /OPR 147 0153 0206 6212 JMP OP14 /YES /OPR 147 0155 0210 0514 OPR 14 /OPR 147 0155 0210 0514 OPR 14 /OPR 147 0156 0211 6130 OTHERS, JMP ERR /PUT JMP TO /CHECKS FOR OTHER 0160 0212 1000 OP14, LOA /OUTPUT, GET CHARA /OUTPUT, GET CHARA 0162 0213 0242	0143	Ø176	625Ø		JMP		OOKBD	/YES.
Ø145 Ø200 Ø535 KBD I Ø146 Ø201 Ø467 SKP /NO, Ø147 Ø202 6250 JMP DOKBO /YES, Ø150 Ø203 1460 SAE I /OPR I 14? Ø151 Ø204 Ø534 OPR I 14 Ø152 Ø205 Ø467 SKP /NO Ø153 Ø206 6212 JMP OP14 /YES Ø155 Ø210 Ø514 OPR 14 Ø155 Ø210 Ø514 OPR 14 Ø156 Ø211 6130 OTHERS, JMP ERR /PUT JMP TO Ø163 Ø212 1Ø00 OP14, LOA /OUTPUT, GET CHARA Ø163 Ø214 Ø500 IOB /INSTRUCTIONS HERE Ø163 Ø214 Ø500 IOB /OUTPUT, GET CHARA Ø164 Ø215 6046 TLSA /OUTPUT IT Ø165 Ø216 Ø500 IOB /CHECK IF DONE Ø166 Ø217 6041 TSFA	0144	0177	146Ø		SAE	I		/IS IT KBO I?
0146 0201 0467 SKP /NO. 0147 0202 6250 JMP DOKBO /YES. 0150 0203 1460 SAE I /OPR I 14? 0151 0204 0534 OPR I 14 /OPR I 14? 0152 0205 0467 SKP /NO 0153 0206 6212 JMP OP14 /YES 0154 0207 1460 SAE I /OPR 14? 0155 0210 0514 OPR 14 /OPR 14? 0156 0211 6130 OTHERS, JMP ERR /PUT JMP TO 0157 //0159 0210 0514 OPR 14 /OUTPUT, GET CHARA 0160 0212 1000 OP14, LOA /OUTPUT, GET CHARA 0161 0212 1000 OP14, LOA /OUTPUT, GET CHARA 0162 0213 0242 AC /OUTPUT, GET CHARA 0163 0214 0500 IOB /OUTPUT IT 0164 0215 6046 TLSA /OUTPUT IT 0165 0216	0145	0200	0535		KBD	I		
0147 0202 0202 0203 JMP DOKBO /VES. 0150 0203 1460 SAE I /OPR I 14? 0151 0204 0534 OPR I 14 0152 0205 0467 SKP /NO 0153 0206 6212 JMP OP14 /YES 0154 0207 1460 SAE I /OPR 14? 0155 0210 0514 OPR 14 0156 0211 6130 OTHERS, JMP ERR /PUT JMP TO 0157 0210 0514 OPR 14 /OTRUCTIONS HERE 0160 0212 1000 OP14, LOA /OUTPUT, GET CHARA 0161 0212 1000 OP14, LOA /OUTPUT, GET CHARA 0163 0214 0500 IOB /OUTPUT, GET CHARA 0164 0215 6046 TLSA /OUTPUT IT 0165 0216 0500 IOB /CHECK IF DONE 0166 0217 6041 TSFA /CHECK IF DONE<	Ø146	0201	0467		SKP			/NO.
0150 0203 1460 SAE 1 70FR I 147 0151 0204 0534 OPR I 14 14 0152 0205 0467 SKP /NO 0153 0206 6212 JMP OP14 YES 0154 0207 1460 SAE I /OPR 147 0155 0210 0514 OPR 14 0156 0211 6130 OTHERS, JMP ERR /PUT JMP TO 0157 //INSTRUCTIONS HERE /INSTRUCTIONS HERE /INSTRUCTIONS HERE 0160 0212 1000 OP14, LOA /OUTPUT, GET CHARA 0162 0213 0242 AC /OUTPUT, GET CHARA 0163 0214 0500 IOB /OUTPUT IT 0164 0215 6046 TLSA /OUTPUT IT 0165 0216 0500 IOB /CHECK IF DONE 0166 0217 6041 TSFA /CHECK IF DONE 0167 0220 6216 JMP ,-2 0170 /RETURN TO USER PROGRAM	014/	0202	0250		JMP		DOKBO	ALES.
0151 0204 0534 UPR 1 14 0152 0205 0467 SKP /NO 0153 0206 6212 JMP 0P14 /YES 0154 0207 1460 SAE I /OPR 14? 0155 0210 0514 OPR 14 0156 0211 6130 OTHERS, JMP ERR /PUT JMP TO 0157 /CHECKS FOR OTHER /INSTRUCTIONS HERE /INSTRUCTIONS HERE 0161 0212 1000 OP14, LOA /OUTPUT, GET CHARA 0162 0213 0242 AC /INSTRUCTIONS HERE 0163 0214 0500 IOB /OUTPUT, GET CHARA 0164 0215 6046 TLSA /OUTPUT IT 0165 0216 0500 IOB /OUTPUT IT 0166 0217 6041 TSFA /CHECK IF DONE 0167 0220 6216 JMP<, -2	0150	0203	1400		SAL	1		70PR I 147
0152 0205 0407 SAF 700 0153 0206 6212 JMP 0P14 YES 0154 0207 1460 SAE I /OPR 147 0155 0210 0514 OPR 14 0156 0211 6130 OTHERS, JMP ERR /PUT JMP TO 0157 //INSTRUCTIONS HERE /INSTRUCTIONS HERE /INSTRUCTIONS HERE 0161 0212 1000 OP14, LOA /OUTPUT, GET CHARA 0162 0213 0242 AC /OUTPUT, GET CHARA 0163 0214 0500 IOB /OUTPUT IT 0164 0215 6046 TLSA /OUTPUT IT 0165 0216 0500 IOB /OHECK IF DONE 0167 0220 6216 JMP ,-2 0170 // // /CHECK IF DONE 0171 /RETURN TO USER PROGRAM /EXIT TRAP PROCESS 0173 0222 0200 OVL,	0151	0204	00004		SKD	1	14	(NO
Ø154 0207 1460 SAE I /OPR 147 Ø155 0210 0514 OPR 14 Ø156 0211 6130 OTHERS, JMP ERR /PUT JMP TO Ø157 0161 0212 1000 OP14, LOA /OUTPUT, GET CHECKS FOR OTHER Ø160 0212 1000 OP14, LOA /OUTPUT, GET CHARA Ø162 0213 0242 AC /OUTPUT, GET CHARA Ø163 0214 0500 IOB /OUTPUT IT Ø165 Ø216 0500 IOB Ø164 0215 6046 TLSA /OUTPUT IT /OHECK IF DONE Ø165 0216 0500 IOB /OHECK IF DONE /OHECK IF DONE Ø166 0217 6041 TSFA /CHECK IF DONE /OHECK Ø167 Ø220 6216 JMP ,-2 /OHECK /OHECK Ø170 / /RETURN TO USER PROGRAM /EXIT TRAP PROCESS Ø173 Ø222 Ø200 OVL, Ø <td>0153</td> <td>0200</td> <td>6212</td> <td></td> <td>IMP</td> <td></td> <td>0014</td> <td></td>	0153	0200	6212		IMP		0014	
0155 0210 0514 OPR 14 0156 0211 6130 OTHERS, JMP ERR /PUT JMP TO 0157 /CHECKS FOR OTHER /INSTRUCTIONS HERE /INSTRUCTIONS HERE 0160 0212 1000 OP14, LOA /OUTPUT, GET CHARA 0162 0213 0242 AC /OUTPUT, GET CHARA 0163 0214 0500 IOB /OUTPUT IT 0164 0215 6046 TLSA /OUTPUT IT 0165 0216 0500 IOB /OHECK IF DONE 0166 0217 6041 TSFA /CHECK IF DONE 0167 0220 6216 JMP ,-2 0170 // // // // 0171 /RETURN TO USER PROGRAM / // 0172 0221 1020 RET, LDA / // 0173 0222 0000 OVL, 0 / /	0154	0200	1460		SAF	T	0114	/ DPR 147
0156 0211 6130 OTHERS, JMP ERR /PUT JMP TO 0157 /CHECKS FOR OTHER 0160 /INSTRUCTIONS HERE 0161 0212 1000 OP14, LOA /OUTPUT, GET CHARA 0162 0213 0242 AC /OUTPUT, GET CHARA 0163 0214 0500 IOB /OUTPUT IT 0164 0215 6046 TLSA /OUTPUT IT 0165 0216 0500 IOB /CHECK IF DONE 0166 0217 6041 TSFA /CHECK IF DONE 0167 0220 6216 JMP ,-2 0170 // /RETURN TO USER PROGRAM 0172 0221 1020 RET, LDA /EXIT TRAP PROCESS 0173 0222 0000 0VL, 0 /	Ø155	0210	0514		OPR	•	14	
0157 /CHECKS FOR OTHER 0160 /INSTRUCTIONS HERE 0161 0212 1000 OP14, LOA /OUTPUT, GET CHARA 0162 0213 0242 AC ////////////////////////////////////	Ø156	Ø211	6130	OTHERS	JMP		ĒRR	ZPUT JMP TO
Ø16Ø /INSTRUCTIONS HERE Ø161 Ø212 1000 OP14, LOA /OUTPUT, GET CHARA Ø162 Ø213 Ø242 AC /000000000000000000000000000000000000	0157						-	/CHECKS FOR OTHER
Ø161 Ø212 1000 OP14, LOA /OUTPUT, GET CHARA Ø162 Ø213 Ø242 AC Ø163 Ø214 Ø500 IOB Ø164 Ø215 6046 TLSA /OUTPUT IT Ø165 Ø216 Ø500 IOB /OUTPUT IT Ø166 Ø217 6041 TSFA /CHECK IF DONE Ø167 Ø220 6216 JMP ,-2 Ø170 / /RETURN TO USER PROGRAM Ø172 Ø221 1020 RET, LDA Ø173 Ø222 Ø000 OVL, Ø	0160							/INSTRUCTIONS HERE,
Ø162 Ø213 Ø242 AC Ø163 Ø214 Ø5ØØ IOB Ø164 Ø215 6046 TLSA /OUTPUT IT Ø165 Ø216 Ø5ØØ IOB /OUTPUT IT Ø166 Ø217 6041 TSFA /CHECK IF DONE Ø167 Ø22Ø 6216 JMP ,-2 Ø170 / /RETURN TO USER PROGRAM Ø172 Ø221 102Ø RET, LDA I /EXIT TRAP PROCESS Ø173 Ø222 ØØØØ OVL, Ø INT INT INT	0161	Ø212	1000	OP14,	LOA			/OUTPUT, GET CHARACTER
Ø163 Ø214 Ø5ØØ IOB Ø164 Ø215 6Ø46 TLSA /OUTPUT IT Ø165 Ø216 Ø5ØØ IOB IOB Ø166 Ø217 6Ø41 TSFA /CHECK IF DONE Ø167 Ø22Ø 6216 JMP ,-2 Ø170 / / ////////////////////////////////////	Ø162	Ø213	0242		AC			
Ø164 Ø215 6046 TLSA /OUTPUT IT Ø165 Ø216 Ø500 IOB IOB Ø166 Ø217 6041 TSFA /CHECK IF DONE Ø167 Ø220 6216 JMP ,-2 Ø170 / / / Ø171 /RETURN TO USER PROGRAM / Ø172 Ø221 1020 RET, LDA Ø173 Ø222 Ø000 OVL, Ø	0163	Ø214	0500		108			
0165 0216 0500 IOB 0166 0217 6041 TSFA /CHECK IF DONE 0167 0220 6216 JMP ,-2 0170 / 0171 /RETURN TO USER PROGRAM 0172 0221 1020 RET, LDA I /EXIT TRAP PROCESS 0173 0222 0000 OVL, 0	0164	0215	6046		TLSA			/OUTPUT IT
0166 0217 6041 TSFA /CHECK IF DONE 0167 0220 6216 JMP2 0170 / 0171 /RETURN TO USER PROGRAM 0172 0221 1020 RET, LDA I /EXIT TRAP PROCESS 0173 0222 0000 OVL, 0	Ø1 65	Ø216	0500		IOB			
0167 0220 6216 JMP ,-2 0170 / 0171 /RETURN TO USER PROGRAM 0172 0221 1020 RET, LDA I /EXIT TRAP PROCESS 0173 0222 0000 OVL, 0	0166	0217	6041		TSFA			/CHECK IF DONE
01/0 / 0171 /RETURN TO USER PROGRAM 0172 0221 1020 RET, LDA I /EXIT TRAP PROCESS 0173 0222 0000 OVL, 0	0167	0220	6216	,	JMP		,-2	
01/1 /RETURN TO USER PROGRAM 0172 0221 1020 RET, LDA I /EXIT TRAP PROCESS 0173 0222 0000 OVL, 0	01/0			/				
0173 0222 0000 OVL, 0	01/1	0004	1 0 0 0	TRETURN	VIOU	SER	PRUGRAM	CALL TOLD DOGGCOOD
	01/2	Ø221	1020	KEI;	LUA	1		ZEALL TRAP PROCESSOR
	0174	0222	2247		Ø		12777	
UIIA UZCO ZZALADU UVN /SIII LAUSES UV	NT/4	0220	6641	AUU	UVN		13111 GA	NUJES UV

e9 6

C

-

1_

Ø175	0224	1020		LDA	Ι		/GET LINK
0176	Ø225	0000	LC,	Ø			
Ø177	Ø226	0321		ROR	I	1	/RESTORE LINK
0200	Ø227	ØØ 4Ø		SET		Ø	/RESTORE LOCATION Ø
0201	Ø23Ø	0017		17			
0202	0231	1020		LDA	1		/RESTORE Q REGISTER
0203	Ø232	0000	Q,	Ø			
0204	0233	0314		ROR		14	ZLOADS O REGISTER
0205						-	VEROM AC
0206	0234	1000					ANDRESS FOR RETURN
0207	1235	0140		140			/TO USED DROCRAM
0207	0205	1600		746			TO OSER PROGRAM
0210	0230	1020		BOF	1		
0211	0231	1246		JMP		0.2.1	
0212	0240	1120		510		RIJ	
0213	0241	1120		AUA	1		TRESTORE AL
0214	0242	0000	AC,	Ø			
0215	Ø243	0006		DJR			/INHIBIT ZERO CLOBBERING
0216	02 44	05 00		IOB			/IN_USERS ME MORY SEGMENT
0217	2245	6244		RMEA			/RESTORE INTERRUPT BUFFER
0220	0246	0000	RTJ,	HLT			/BECOMES JMP TO USER PROG.
0221	Ø247	3777	OVN,	3777			
0222			/				
0223			/KBD PR	OCESS	ING		
0224	Ø25Ø	0325	DOKBD,	ROR	I	5	/PUT TRAPPED INSTRUCTION
Ø225							/I BIT INTO COMPUTER
0226							/LINC BIT
0227	Ø251	1020		LDA	I		/GET LEFTOVER CHAR,
0230	0252	0000	UPC,	ø			/IF ANY
Ø231	Ø253	0450		AZE			/LEFTOVER CHARACTER?
0232	Ø254	6323		JMP		100	YES, DELIVER IT TO USER
0233	0255	0435	LP.	KST	I		/NO. FRESH CHARACTER?
0234	0256	6262		IMP	-	GRC	YES. GO READ IT
0235	0257	0452		17F		0.10	/NO PAUSE?
2236	0260	6255		IMP		IP	YES HANG IN THERE
0237	5200			0111		1 -1	TTUL KEY STRUCK
0207	0261	6323		IMD		1.00	
0241	0201	0020		Jun		200	ACLEADED
0242			1				CLEARED
0243			APEAD AN				
0240	1262	12 12 1 1	CDC	יאו טאי	A IN DI	LAICAN	LIBUARU CHARACIER
0274	0202	0011	GRUI				
8044	0200	6070		108			
0240	0204	1100		KRSA			TREAD A CHARACIER
024/	0205	1120		AUA	1		/CALCULATE CHARACIER CON-
0250	0200	/602		-1/5			/VERSION TABLE INDEX IN
0251	0267	0301		ROR	_	1	/HALF WORDS
0252	0270	1560		BCL	I		
0253	0271	3700		3700			
0254	0272	1120		ADA	I		/ADD BASE ADDRESS OF TABLE
0255	0273	0401		TAB			
Ø256	0274	4276		STC		.+2	/HALF-WORD PICKUP ADDRESS
Ø257	Ø275	1300		LDH			/GET THE LINC CHARACTER,
0260	Ø276	ØØØØ		Ø			
0261	Ø277	1420		SHD	I		/IS IT AN UNDEFINABLE?
0262	0300	7600		7600			
0263	0301	6337		JMP		RETX	/YES, ECHO UPARROW
0264	0302	1120		ADA	I		/IS IT UPPER CASE?
0265	0303	7722		-55			
0266	0304	0451		APO			

- -

6 S

0304 0451 0305 6317

APO JMP LOW /NC. / /ITS AN UPPER CASE CHARACTER ADA I /MAKE IT LINC CODE 11 0267 0270 0271 0272 0273 0306 1120 0307 0011

0451 /DELETE MINUS ZEROES APO 0274 0310 0011 Ø275 Ø311 CLR /SAVE IT FOR USE AFTER /CASE CHARACTER IS OUTPUT UPC Ø276 Ø312 4252 STC 0277 /GET CASE CHARACTER Ø313 1020 LDA I 0300 23 0314 0023 0301 0315 STC AC /STORE FOR TRANSFER 4242 0302 RET /EXIT 0303 Ø316 6221 JMP 0304 /LOWER CASE CHARACTERS 0305 Ø317 LOW, ADA I /MAKE IT LINC CODE 1120 0306 55 0055 0320 0307 APO 0451 0310 Ø321 Ø311 Ø322 0011 CLR /STORE IT FOR TRANSFER Ø323 4242 L00, STC AC 0312 /CLEAR LEFT OVER BUFFER 0324 4252 STC UPC 0313 0325 0500 108 0314 /READ CHAR AND 0315 0326 6036 KRBA /CLEAR KEYBOARD FLAG 0316 0327 635Ø L001, JMP PRINT /ECH0 0317 1460 SAE /C, R.? Ø33Ø Ι 0320 215 0331 Ø215 0321 JMP RET /NO, EXIT Ø332 6221 0322 /YES. THROW IN 1020 LDA I 0323 0333 /A FREE LINE FEED 0324 0334 0212 212 0335 635Ø JMP PRINT 0325 JMP Ø326 Ø336 6221 RET Ø327 /ILLEGAL CHARACTER 0330 LDA I /PRINT UP ARROW 0337 1020 0331 RETX, 0332 0340 0336 336 0341 6350 JMP PRINT 0333 Ø342 Ø5ØØ IOB /ECHO OFFENDING 0334 0343 6036 KRBA /CHARACTER. 0335 Ø344 PRINT JMP 0336 635Ø Ø345 0337 0017 COM /SET LINK BIT 0340 0346 0261 ROL I 1 /TO KEYBOARD "PAUSE" VALUE 0341 JMP 0347 6255 LP 0342 0343 1 /PRINT A CHARACTER 0344 /ENTER W ASCII CHAR IN AC Ø345 PRINT, PDP 0346 Ø35Ø 0002 0347 PMODE 0350 4351 6046 TLS 0351 4352 6041 TSF 0352 4353 5352 JMP .-1 LINC Ø353 4354 6141 0354 LMODE 0355 0355 6000 JMP Ø /INITIALIZE THE TRAP PROCESSOR 0356 0357 0356 0011 INIT: CLR 0357 4252 STC UPC /CLEAR LEFT OVER CHARACTER 0360 /BOILER PLATING FOR GUIDE 0360 0001 AXO 0361 /RESTART 0362 /GET SPEC, FNS. REG. /SET TRAP ENABLE BIT 0024 SFA 0363 0361 0362 BSE 0364 1620 I 1000 1000 0365 0363 /READ IT BACK OUT 0004 FSF 0364 0366 /RETURN JMP ENDX, Ø 0367 0365 6000 0370 *400 2400 6025 JMP RST400 /GO TO START 400 0371 /CONTROL ROUTINE 0372

14

.

->

			,			
3373			1			
03/4	4401	2423	7 4 9	2223		E-CASE / AL THODE-CASE
03/2	0401	1776	IAD;	1376		
0376	0402	13/0		7676		
03//	0403	7070		7070		20
0400	0404	/0/0		7670		
0401	0405	/0/0		/0/0		П
0402	0400	/0/0		/0/0	/BELL/N	
0403	0407	/050		7000	/TAB/L.	F. THEIA
0404	6410	/0/0		/0/0	/N. U.	
0405	0411	1276		1270	TUR-EUL	/N. U.
0406	Ø412	1676		/6/6	/N. U.	
3407	0413	7676		7676	/N, U.	
0410	0414	7676		/676	/N. U.	
0411	0415	7676		7676	/N. U.	
0412	0416	7676		7676	/N. U.	
0413	Ø417	7676		76 7 6	/N. U.	
0414	0420	7676		7676	/N. U.	
0415	0421	2376		2376	/ AL TMOD	E-CASE/N, U,
0416	0422	7614		7614	/N.U./S	PACE-SPACE
0417	0423	7676		7676	/EXC.PI	/OBL,QUOTES
0420	0424	2265		2265	/NUMBEF	SIGN/OOLLAR SIGN-
0421	Ø425	6215		6215	/PERCEN	T-P, C/AMPERSAMO-AMPERSANU
0422	0426	1676		1676	/APOSTF	OPHE-AP, /OP PAREN
0423	Ø427	7676		7676	/CL PAF	EM/*
0424	Ø43Ø	2063		2063	/PLUS-4	/COMMA-,
0425	0431	1764		1764	/MINUS-	-/PERIOD
0426	0432	2100		2100	/SLASH ·	SLASH/ZERO-Ø
Ø427	0433	0102		0102	/1/2	
0430	0434	0304		0304	/3/4	
0431	0435	0506		0506	/5/6	
0432	0436	0710		0710	/7/8	
0433	Ø437	1176		1176	/9/N.U.	
0434	0440	7676		7676	/SEMICO	LON/
0435	0441	6176		6176	/EQUAL	SIGN-=/>
0436	0442	7676		7676	/?/AT S	IGN
0437	Ø443	2425		2425	/A/B	
0440	0444	2627		2627	/c/D	
Ø441	0445	3Ø31		3031	/E/F	
0442	0446	3233		3233	/G/H	
0443	2447	3435		3435	/1/J	
0444	6450	3637		3637	/K/L	
0445	0451	4041		4041	/M/N	
0446	\$452	4243		4243	/0/P	
0447	\$453	4445		4445	/Q/R	
0450	0454	4647		4647	/S/T	
0451	0455	5051		5051	/U/V	
0452	Ø456	5253		5253	/W/X	
0453	0457	5455		5455	1412	
0454	0460	6621		6621	/LEFT.E	RACK/BACK SLASH
Ø455	0461	7676		7676	/RT BR/	CK/UP ARROW
0456	0462	7676	TABE,	7676	/BACK /	RROW/N.U.
0457			/			•
0460			/INITIAL	START	20	
0461	0463	6474	ST20.	JMP	SETUP	
0462	0464	6350		JMP	INIT	
0463	0465	6472		IMP	GOMAN	
0464	0,00	0.72	/	Q m	GOTAN	
0465				START	400	
0466	0466	6474	ST400	IMP	SETUP	
0467	0467	6.356	1001	IMP	INTT	VINITIALIZE TRAP PROCESSOR
0470	(2 A 7 (A	06030			3	A THEFT IN THOUSON
04/0	0470	0043			5	
		17 C M C C C C C C C				

.

ن

0472	0472	0600	GOMAN,	LIF		ø	
0473	0473	6033		JMP		GUIDE	/THIS LANDS IN FIELD Ø
0474			TRAP PF	OCESS	SOR	RELOCAT	OR
0475	0474	0057	SETUP,	SET		17	
0476	Ø475	0000		Ø			
0477	0476	1020		LDA	I	/SET UP	RESTARTS
0500	Ø477	6356		JMP		INIT	
0501	0500	1040		STA			/SET UP 20 RESTART
0502	0501	0020		20			
0503	0502	4025		STC		RST400	/SET UP 400 RESTART
0504	0503	0640		LDF		Ø	/SET DATA FIELD TO Ø
0505	0504	0061		SET	I	1	/SET UP MOVE COUNT
0506	0505	7334		-TABE	:+17	7	
0507	0506	0062		SET	I	2	/SET TO-ADDRESS
0510	0507	2017		2017			/BIT 1 (2-010) SAYS USE DF
0511	0510	0063		SET	I	3	/SET FROM-ADDRESS
Ø512	0511	0017		17			/BIT 1 (0-000) SAYS USE IF
0513	0512	1023		LDA	I	3	/INCREM, 3,GET (3)
0514	0513	1062		STA	I	2	/INCREM,2,STORE (2)
0515	0514	0221		XSK	I	1	/INCREM, 1, SKIP IF 1=1777
0516	0515	6512		JMP		3	/1 NOT 1777. LOOP AGAIN
0517	0516	0062		SET	I	2	/SET UP FOR LEFT
Ø52Ø	Ø517	2677		2677			/SWITCHES RESTART
0521	0520	0061		SET	I	1	/20 JMP GUIDE-S
Ø522	Ø521	7757		-20			
0523	0522	1020		LDA	1		
0524	0523	6033		JMP		GUIDE	
0525	0524	1062		STA	I	2	/INC 2, STORE INTO (2)
0526	0525	0221		XSK	I	1	/END CHECK
0527	0526	6524		JMP		2	
0530	Ø527	6017		JMP		17	

86 F L Q.

C

0000 ERRORS

AC CDFX	4242 4171
DOKBD	425Ø
ENDX	4365
ERR	4130
FETCH	4160
GOMAN	4472
GRC	4262
GUIDE	4033
INIT	4356
KBD	Ø515
KCCA	6032
KRBA	6036
KRSA	6034
LC	4225
LOO	4323
L001	4327
LOW	4317
LP	4255
M7Ø	4135
ONE	4136
OPR	0500
0P14	4212
OTHERS	4211
OVL	4222
UVN	424/
PC	4134
PRINT	4070
U	4232
40	

ĩ._.:

and a general

RET	4221
RETX	4337
RMFA	6244
RST400	4025
RTJ	4246
SETUP	4474
ST2Ø	4463
ST400	4466
TAB	4401
TABE	4462
TLSA	6046
TSFA	6041
UPC	4252

.