

PDP-12 LIBRARY

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 O Ç

CONTENTS

.

•

C

C

		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)

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	

iv

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.

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,

ILO LESIMOD)

(The symbol I 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
	(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.)

	TRAPPED OPRS.	
PLOT,	STA 1 Ø	ZSAVE INSTRUCTION
PLINST,	BCL I	/IGNORE INSTRUCTION 1 BI
		I HONORE INSTRUCTION I BE
	ADA I	
	-0PR-12	
	APO I	/OPR 12 OR LESS?
	JMP PLERR	/NO• ERROR
	ADA I	YYES. OK SO FAR
	6	
	APO	/OPR 5 OR MORE?
	JMP PLERR	ZNO. ERROR
	ADA I	/YES. OBTAIN COMMAND
	ADD PLTAB-1	
	STC •+1	
	HI T	ZBECOMES ADD PLTAB + N
	STC PIGO	
	PDP	
	PMODE	
	PLCF	
PL 60,	HLT	ZPIPU, PIPK, ETC
	PLSF	
	JMP1	
	L. I NC	
	LMODE	1
	JMP RET	/RETURN TO USER
/		Υ.
	IN NOT RECOGNIZED	
PLERR,	L.DA	
	PLINST	
	JMP ERR	/DO ERROR STOP
	PDP	
	PMODE	
PLTAB,	PLPD	/PEN DOWN. OPR 5
	PLPU	ZPEN UP
	PLDU	ZDRUM DOWN
	PLDD	ZDRUM UP
	PIPK	ZPEN RIGHT
	PIPI	ZPEN LEFT .OPR 12
	I I NG	
	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:

GUTDE. J	JMP INIT	ZINITIALIZE TRAP PROCESSOR				
	LDF 2					
	RDC	ZREAD 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	2	ZPUT RCG 7300 IN
	RCG Sta		/4016 AND 4017
	2016 I DA	I	
	7300 Sta		
	2017		
	LDF LIF	3 2	
	JMP	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:

] (10),	SIC	AC	ISTORE IT FOR TRANSFER
	STC	UPC	ZCIFAR LEFT OVER BUFFER
	I OB		
	KRBA		ZREAD CHAR AND
			ZCLEAR KEYBOARD FLAG
1 ()(1] ,	JMP	RF T	/RETURN TO USFR PROGRAM
1			
/ILLEGAL C	CHARACTER		
RETX,	LDA T		/PRINT UP ARROW.
	٠		
	•		
	•		
	•		

*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.

+++~	1	Ω	G	1	1.	\mathcal{O}	- 1	\sim J	40	۰ 1	nu -
NM	4	N	4	5-	0	·O	° 2	$\langle O \rangle$	4	4	n.
()	(N	"Q	\sim	4	4	1	V.	4	۳2	23	(NI
	Ŷ	4	4	4	4	47	4	4	Q	Ś	1
a stand		5									
		2		۵.		3					
· ×	4	4		\supset	63	3		w	4	∢	
pro pro											\cup
لما لما											
αa	a	Ω.	α	S	ŝ	S	ş	۶	*	-	2

C

	1010					
AC	4242					
CDFX	4171					
DOKBD	4250					
ENDX	4365					
ERR	4130					
FETCH	4162					
GOMAN	4472					
GRC	4262					
GUIDE	4033					
INIT	4356					
KBD	Ø 51 5					
KCCA	6032					
KRBA	60 36					
KRSA	6034					
KR5A	400-					
LĆ	4225					
L00	4323					
L001	4327					
LOW	4317					
LOW LP	4255					
M7Ø	4135				1	
ONE	4136					
OPR	0500			`		
0P14	4212		×			
OTHERS	4211					
OVL	4222					
OVN	4247					-
PC	4134					
PRINT	4350					
Q	4232					
9	72.02					

Q

O

۲



-~

•

C

C

0472	0472	0600	GOMAN,	LIF		Ø	
0473	2473	6033		JMP		GUIDE	/THIS LANDS IN FIELD Ø
0474	-		/TRAP PI	ROCESS	SOR	RELOCATI	
0475	2474	ØØ57	SETUP,			17	
0476	2475	8202		Ø			
0477	2476	1020		LDA	I	/SET UP	RESTARTS
0500	6477	6356		JMP		INIT	
0501	2500	1040		STA			/SET UP 20 RESTART
0502	0001	6328		20			
0503	6502	4025		STC		RST400	/SET UP 400 RESTART
0504	2523	0640		LDF		Ø	/SET DATA FIELD TO Ø
0505	2504	0261		SET	I	1	/SET UP MOVE COUNT
0506	0505	7334		- TABE	+1	7	
0507	0506	0062		SET	I	2	/SET TO-ADDRESS
0510	0507	2017		2017			/BIT 1 (2=010) SAYS USE DF
Ø511	Ø51Ø	6663		SET	Ι	3	/SET FROM=ADDRESS
Ø512	\$511	2017		17			/BIT 1 (0-000) SAYS USE IF
0513	0512	1023		LDA	I	3	/INCREM, 3,GET (3)
Ø514	Ø513	1062		STA	I	2	/INCREM,2,STORE (2)
Ø515	0514	0221		XSK	I	1	/INCREM, 1, SKIP IF 1=1777
Ø516	0515	6512		JMP		,-3	/1 NOT 1777. LOOP AGAIN
0517	0516	0062		SET	I	2	/SET UP FOR LEFT
0520	2517	2677		2677			/SWITCHES RESTART
0521	Ø52Ø	0061		SET	I	1	/20 JMP GUIDE-S
ø522	Ø521	7757		-2ø			
0523	0522	1020		LDA	I		
Ø524	Ø523	6033		JMP		GUIDE	
Ø525	0524	1062		STA	I	2	/INC 2. STORE INTO (2)
Ø526	Ø525	Ø221		XSK	I	1	/END CHECK
0527	2526	6524		JMP		,-2	~
Ø53Ø	0527	6Ø17		JMP		17	

\$

Ø373			1	\square
Ø374 🛀		0107	/ TAB, 2323	/ALTMODE-CASE/ALTMODE-CASE
Ø375 Ø376	2401 0402	2323	TAB, 2323 1376	/RUBOUT-DEL/LEADER-NULL
Ø377	2473		7676	/NOT USED
0400	(14)14 (14)14	7676	7676	/N, U.
0401	2425	7075	7676	/N. U.
0402	_ 4.° o	7070	7676	/BELL/N. U.
0403	. - د	ెందం	7656	/TAB/L. FMETA
0404	1 č	్రౌ 5	7675	/N. J.
0405	2411	1270	1276	/CR-EOL/N. U.
0406	1412	7075	7676	/N. U.
0407	2413	7576 76 7 5	7676 7676	/N. U. /N. U.
Ø410 Ø411	2414 2415	00 7075	7676	/N. U.
Ø412	2416	7676	7676	/N. U.
0413	2417	7676	7676	/N. U.
0414	2422	7676	7676	/N, U.
0415	2421	2376	2376	/ALTMODE-CASE/N, U.
0416	2422	7614	7614	/N.U./SPACE-SPACE
Ø417	2423	7676	7676	/EXC.PT/DBL.QUOTES
0420	2424	2255	2265	/NUMBER SIGN/DOLLAR SIGN-
0421	2425	6215	6215	/PERCENT-P.C/AMPERSAMD-AMPERSAND
0422	2425	1676	1676	/APOSTROPHE=AP,/OP PAREN
0423	2427	7676	7676	/CL PAREM/* /PLUS-+/COMMA-,
Ø424	2432 2431	2263 1764	2063 1764	/MINUS/PERIOD
Ø425 Ø426	2432	2122	2100	/SLASH-SLASH/ZERO-Ø
0427	2433	2122	0102	/1/2
0430	2434	2324	0304	/3/4
0431	2435	2526	0506	/5/6
Ø432	2435	2712	Ø71Ø	/7/8
0433	2437	1176	1176	/9/N.U.
0434	2442	7576	7676	/SEMICOLON/<
Ø435	2441	6175	6176	/EQUAL SIGN-=/>
0436	2442	7676	7676	/?/AT SIGN
0437	2443	2425	2425	/A/B
0440	2444	2627	2627	/ C/ D / E / F
Ø441 Ø442	2445 2446	3231 3233	3Ø31 3233	/ E / F / G / H
Ø442 Ø443	2447	3435	3435	
0444	2452	3537	3637	/K/L
0445	2451	4241	4041	/M/N
0446	, 452	4243	4243	/0/P
0447	2453	4443	4445	/Q/R
0450	₹454		4647	/S/T
0451	2455	5251	5051	/U/V
0452	2456	5253	5253	
0453	2457	5455	5455	
0454	2462	6621	6621 7676	/LEFT.BRACK/BACK SLASH /rt.brack/up_arrow
Ø455 Ø456	2461	7676 7676	TABE, 7676	ZBACK ARROW/N.U.
0457	1422	10 0	/ / / / / / / / / / / / / / / / / / /	Brok Hudowy N. O.
0457 0460			/INITIAL START	20
0461	2463	5474	ST20, JMP	SETUP
Ø462	2404	635c	JMP	INIT
0463	2465	6472	JMP	GOMAN
Ø464			1	
Ø465			/INITIAL START	
0466	2466	0474	57400, JMP	SETUP
6467	2467	5356	Jwb	INIT /INITIALIZE TRAF PROCESSOR

ŧ

0274	0310	0451		APO			/DELETE MINUS ZEROES
0275	Ø311	0011		CLR			
0276	0312	4252		STC		UPC	/SAVE IT FOR USE AFTER
0277							/CASE CHARACTER IS OUTPUT
0300	2313	1222		LDA	I		/GET CASE CHARACTER
0301	2314	2223		23			
0302	2315	4242		STC		AC	/STORE FOR TRANSFER
0303	2316	6221		JMP		RET	/EXIT
0304			1				
0305			/LOWER	CASE CH	HAR	ACTERS	
0306	2317	1120	LOW,	ADA	I		/MAKE IT LINC CODE
0307	0320	0055		55			
0310	0321	0451		APO			
Ø311	0322	2011		CLR			
0312	2323	4242	L00,	STC		AC	STORE IT FOR TRANSFER
0313	£324	4252		STC		UPC	/CLEAR LEFT OVER BUFFER
Ø314	Ø325	0500		IOB			
Ø315	@326	6036		KRBA			ZREAD CHAR AND
0316							/CLEAR KEYBOARD FLAG
0317	0327	6350	L001,	JMP	-	PRINT	/ECH0
0320	0330	1460			I		/C, R.?
0321	0331	0215		215			
0322	2332	6221		JMP	•	RET	/NO, EXIT /Yes, Throw in
0323	2333	1023			I		A FREE LINE FEED
0324	0334			212			VA FREE LINE FEED
0325	0335	635Ø		JMP		PRINT RET	· · · · · · · · · · · · · · · · · · ·
Ø326	Ø336	6221	,	JMP			
Ø327			/	AL CHAR	A C T	FD	
Ø33Ø Ø331	0337	1020	RETX,	LDA			UP ARROW
0332	2340	0336		336	+		of Annon
0333	2341	6352		JMP		PRINT	
Ø334	0342	0500		IOB		, ,, ,, ,, ,	/ECHO OFFENDING
0335	0343	6036		KRBA			/CHARACTER.
Ø336	0344			JMP		PRINT	
0337	Ø345			COM			
0340	0346	Ø261		ROL	I	1	/SET LINK BIT
0341							/TO KEYBOARD "PAUSE" VALUE
0342	0347	6255		JMP	`	LP	
0343			1				
0344				A CHAR			
Ø345				W ASCI	I C	HAR IN	AC
0346	Ø35Ø	0002	PRINT,	PDP			
0347				PMODE			
0350	4351	6046		TLS			
Ø351	4352	6041		TSF			
0352	4353	5352		JMP		1	
0353	4354	6141		LINC			
Ø354 Ø355	0355	6000		LMODE JMP		Ø	,
0355	8322	0000			45		ROCESSOR
Ø356 Ø357	0300	0011	INIT,	CLR	1	111.41	
0360	0357	4252	TATIR	STC		UPC	/CLEAR LEFT OVER CHARACTER
Ø361	0360	0001		AXO		0.0	/BOILER PLATING FOR GUIDE
0362	0000	6 C. C. 4		47.0			/RESTART
Ø363	2361	0024		SFA			/GET SPEC, FNS, REG,
Ø364	2362 2362	1622			I		SET TRAP ENABLE BIT
0365	0363	1000		1000	-		
Ø366	Ø364	0004		ESF			/READ IT BACK OUT
0367	A365	6000	ENDX,	JMP		Ø	/RETURN
0370	\mathbf{V}^{-}		+400				\bigcirc
0371	0400	6025		JMP		RST400	/GO TO START 400
0372							/CONTROL ROUTINE
aninestation in the little states of				_			

í

.

2120 2227 2020 227 2040 SET 2 ARESTORE LOCATION 8 2221 2237 2040 SET 2 ARESTORE LOCATION 8 2222 2237 2040 0. 8 /RESTORE LOCATION 8 2222 2237 2040 0. 8 //RESTORE COLLECTION 8 2232 2040 0. 8 //RESTORE COLLECTION 8 2233 2311 122 1/4 //RESTORE COLLECTION 8 2234 2233 2314 1/4 //DETTER 2235 2234 122 1/4 //DETTER //DETTER 2237 2240 122 1/4 //DETTER //DETTER 2237 2243 122 1/4 //DETTER //DETTER 2237 2244 2400 STO RTJ //DETTER //DETTER 2240 2247 2247 2248 AUD 1/DETTER //DETTER 2241 2244 2240 2252 0040 1/DETTER //LNC BIT 2242 2252 2040	2175	1224	1020		LDA	I		/GET LINK
2177 2262 227 2040 SET 2 /RESTORE LINK, / 2262 2237 2040 SET 2 /RESTORE LOCATION 0 2262 2231 2020 LDA : /RESTORE LOCATION 0 2262 2231 2020 LDA : /RESTORE CORDISTER 2203 2232 0000 D, 2 2204 2233 2314 ROR 14 /LOADS 0 REGISTER 2205 //ROM AC 2206 2244 1020 LDA / /DETAIN ADDRESS FOR RETURN 2217 2255 0147 140 //DUSER PROGRAM 2210 2256 0147 140 //DUSER PROGRAM 2211 2237 6000 AC, 0 2213 2244 946 STC RTJ 2213 2244 946 STC RTJ 2215 2243 2000 AC, 0 2215 2243 2000 AC, 0 2215 2243 2000 AC, 0 2215 2243 2000 AC, 0 2216 2244 1120 AJA I 2217 2245 6244 RMFA /RESTORE AC 2220 //KHO PROCESSING 2221 2247 3777 OVN, 3777 2222 //KHO PROCESSING 2222 //KHO PROCESSING 2223 //KHO PROCESSING 2224 2247 3777 OVN, 3777 2225 //KHO PROCESSING 2226 //KHO PROCESSING 2227 2251 1020 LDA I /GET LEFTOVER CHAR, 2238 2450 200 UPC, 0 2247 2777 04N, 3777 2258 2452 200 UPC 0 2259 //LINC BIT INTO COMPUTER 2250 //LINC BIT 2250 //LINC BIT 2251 2250 4352 LP, KST I /NO, FRESH CHARACTER? 2253 2257 0452 L2E /NO, PAUSER IT TO USER H. AC 2258 2458 ASE AE 2259 2452 LP, KST I /NO, FRESH CHARACTER? 2259 2452 JMP LOO /YES, GO READ IT 2350 2257 0452 L2E /NO, PAUSE? MARACTER? 2351 2258 2452 JMP L00 /NO, RETURIT TO USER H. AC 2259 //LINC BIT 2350 2257 0452 L2E /NO, PAUSE? AREAD IT 2350 2257 0452 L2E /NO, PAUSE? 2360 2257 0452 L2E /NO, RETURIT TO USER H. AC 2264 //LEA AND TRANSLATE A KEYBOARD CHARACTER 2370 //READ AND TRANSLATE A KEYBOARD CHARACTER 2447 2260 1021 //READ AND TRANSLATE A KEYBOARD CHARACTER 2459 2253 277 1560 BCL I 2550 277 3361 ROR 1 /HALF WORDS 2567 277 3361 ROR 1 /HALF WORDS 2577 277 1300 3722 2584 4272 1120 ADA I /ADD BASE ADDRESS OF TABLE 2585 277 3701 3722 2597 277 1300 LDH //SET THE LINC CHARACTER 2450 2667 752 758 2577 277 1300 LDH //SET THE LINC CHARACTER 2450 2667 752 758 2577 277 1300 LDH //SET THE LINC CHARACTER 2560 233 7722 -55 2577 277 1424 SHO I /IS IT UPPER CASE? 2561 2370 3772 -55 2577 277 1424 SHO I /IS IT AN UNDEFINABLE? 2560 233 7722 -55 2577 277 1424 SHO I /IS IT UPPER CAS				LC,				
222: 223: 224: 225: 225: 225: 225: 225: 225: 225: 225: 225: 225: 225: 225: 225: 225: 225: 225: 225: 225:		3 3226	3321			1		
2222 2231 1220 LDA 1 /RESTORE 0 REGISTER 2233 2333 2314 ROH 14 /LOADS 0 REGISTER 2246 2233 2314 ROH 14 /OBTAIN ADDRESS FOR RETURN 2247 2355 1620 BSE I /OBTAIN ADDRESS FOR RETURN 2213 2234 1320 LDA /OBTAIN ADDRESS FOR RETURN 2214 2235 1620 BSE I /OBTAIN ADDRESS FOR RETURN 2213 2244 2404 4246 STC RTJ 2214 2247 2006 DUB /INUSERS FMEMORY SEGMENT 2215 2244 2006 DUG //INUSERS FMEMORY SEGMENT 2224 2247 3777 OVN, 3777 //INUSERS FMEMORY SEGMENT 2225 2247 3777 OVAB, ROR I //INTID COMPUTER 2226 2247 3777 OVAB, ROR I //INTID COMPUTER 2227 2228 2248 2257 A252 //INTID COMPUTER 2228 2257 2325 2458 JMP LOU	0202		0040		SET		2	/RESTORE LOCATION Ø
223 2232 3000 0 0 224 2233 3314 ROR 14 //LOADS 0 REGISTER 226 2234 1232 LDA /OBTAIN ADDRESS FOR RETURN 221 2235 1220 LDA /OBTAIN ADDRESS FOR RETURN 2212 2235 1620 BSE I 2213 2244 120 ADA I /RESTORE AC 2214 2244 2280 ADA I /RESTORE AC 2215 2244 2000 LDB /IN USERS MEMORY SEGMENT 2216 2244 2000 RU, HIT /RESTORE AC 2221 2244 2000 RU, HIT /RESTORE AC 2222 2244 2000 RU, HIT /RESTORE AC 2223 2244 2000 RU, HIT /RESTORE AC 2224 2257 2325 DOKAD, RDR I 5 /PUT TRAPPED INSTRUCTION 2225 2244 2252 2250 DOKAD, RDR I 5 /LETOVER CHARACTER? 2233 2254 6323 JPP LDA /LETOVER CHARACTER? 224<								
2233 2314 RDH 14 /LDADS Q REGISTER 2265 2234 1203 LDA /OBTAIN ADDRESS FOR RETURN 2265 2234 1203 LDA /OBTAIN ADDRESS FOR RETURN 2267 2235 0140 140 /TO USER PROGRAM 2210 2234 1200 JUP /TO USER PROGRAM 2211 2234 2404 ATA I /RESTORE AC 2213 2244 2406 D.R /INH IBIT ZERO CLOBBERING 2214 2244 2426 IDB /IN USERS MEMORY SEGMENT 2217 2244 2426 IDB /IN USERS MEMORY SEGMENT 2221 2244 2408 RTJ, HLT /RESTORE INTERPED 2222 2446 2408 RTJ, HLT /RECOMES JMP TO USER PROGR 2223 2446 2408 RTJ, HLT /RECOMES JMP TO USER PROGR 2224 2452 3252 00KBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2225 //KBD PROCESSING //LOC OIT /VESTORE CHARACTER 2231 2254 4252 0006 /READ <td></td> <td></td> <td>1020</td> <td></td> <td></td> <td>I</td> <td></td> <td>RESTORE Q REGISTER</td>			1020			I		RESTORE Q REGISTER
2255 2234 1222 LDA //FROM AC 2265 2235 0142 142 /TO USER PROGRAM 2212 2235 0142 142 /TO USER PROGRAM 2212 2244 1220 BSE I //YESTORE AC 2213 2244 120 ADA I /RESTORE AC 2214 2244 246 STC RTJ 2215 2244 2600 ADA I /RESTORE AC 2214 2244 2600 ADA I /RESTORE AC 2214 2244 2600 RTJ, HIT /RESTORE INFERNET 2216 2244 2624 2624 RMFA /ARESTORE INFERNET 2220 2246 2000 RTJ, HIT /SECOMES JMP TO USER PROC. 2221 2247 277 DVN, 3777 /LINC BIT /LINC BIT 2222 2241 3252 2325 DOKBD, ROR I 5 /LINC BIT /LINC BIT 2223 2245 2325 2452 LDA I /GET LEFTOVER CHAR, //E 2242 225				Q,				
Desc. 2234 1202 LDA /OBTAIN ADDRESS FOR RETURN 0207 2235 0149 140 /TO USER PROGRAM 0210 2235 0149 140 /TO USER PROGRAM 0211 2235 0149 JAP /TO USER PROGRAM 0212 2244 122 0402 ASE I 0213 2244 1240 ASA I /RESTORE AC 0214 2244 2440 ASA I /RESTORE AC 0214 2444 2440 ASE //INI USERS MEMORY SEGMENT 0217 2245 2244 2460 DAR /INI USERS MEMORY SEGMENT 0221 0244 2460 DAR //INI USERS MEMORY SEGMENT 0222 0246 2430 RCR FOR TRAPPED INSTRUCTION 0222 02450 3250 00KBD, ROR IS //VIT TRAPPED INSTRUCTION 0222 2254 3257 0325 DOKBD, ROR IS //VIT SEGMENTER 2231 2257 2452<		2233	2314		ROR		14	
0207 C235 0140 140 /TO USER PROGRAM 0210 C236 1620 BSE I JMP 0211 C237 6020 JMP 0212 C240 4246 STC RTJ 0213 C241 1120 AJA I /RESTORE AC 0214 C242 2000 AC, 0 //INHIBIT ZERO CLOBBERING 0215 C244 2000 AC, 0 //INHIBIT ZERO CLOBBERING 0216 C244 4500 DJR /INHIBIT ZERO CLOBBERING 0221 C244 4500 RTJ, 4LT /RESTORE INTERPY SEGMENT 0222 0246 0000 RTJ, 4LT //YED //RESTORE INTERPY SEGMENT 0223 0246 0000 RTJ, 4LT //YED //RESTORE INTERPY SEGMENT 0224 2252 0245 00KBD, ROR I //YED //YED 0225 //YTT OVN, 3777 OVN, 3777 //YES //YES //YES 0225 00KBD, ROR I //YES //YES //YES //YES 0224 2257 0252 JMP LOO /YES<			1001					
2210 2236 1428 BSE I 2211 2237 6203 JPP 2212 2241 1120 AC C 2213 2241 1120 AC C 2214 2442 4260 AC C 2215 2243 2400 AC C 2217 2245 6244 AFA /RESTORE AC 2217 2245 6244 AFA /RESTORE INTERNUT BUFFER 2217 2245 6244 AFA /RESTORE INTERNUT BUFFER 2221 2247 3777 DVN, 3777 2222 //KBO PROCESSING //LEFTOVER CHAR, /LEFTOVER CHAR, 2224 /252 4325 DOKBD, ROR I /LEFTOVER CHAR, 2226 //KBO PROCESSING /LEFTOVER CHAR, /LEFTOVER CHAR, 2226 //S26 4252 JPP LD //I BIT INTO COMPUTER 2233 2254 6323 JPP LO //I CEST LEFTOVER CHAR, 2234 2256 6262 JPP KEST NO, FRESH CHARACTER?								
2211 237 6000 JAP 2212 2244 1120 ADA I /RESTORE AC 2214 2242 0200 AC, 0 2215 2243 2240 0200 AC, 0 2216 2244 0400 D,R /INHIBIT ZERO CLOBBERINC 2216 2244 0400 D,R /INHIBIT ZERO CLOBBERINC 2221 2244 0400 RTJ, HLT /RESTORE INTERRUCTION 2222 2244 0400 RTJ, HLT /RECOMES JMP TO USER PROC. 2221 2244 0200 RTJ, HLT /RECOMES JMP TO USER PROC. 2222 2244 0200 RTJ, HLT /RECOMES JMP TO USER PROC. 2223 2254 0325 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2224 2252 0200 UPC, Ø /IETOVER CHARACTER? 2232 2254 0325 DOKBD, ROR I 5 /LINC BIT 2233 2254 0200 UPC, Ø /IETOVER CHARACTER? 2234 2256 0432 JPP LDO /VES, NEEH CHARACTER?								TU USER PRUGRAM
0212 1244 4246 STC RTJ 0213 2241 1120 ADA /RESTORE AC 0214 2242 0200 AC, 0 //INUSERS MEMORY SEGMENT 0215 2243 0200 DJR /INUSERS MEMORY SEGMENT 0217 2245 6244 WEA /RESTORE INTERPTED SUFFER 0221 2245 6244 WEA /RESTORE INTERPTER 0221 2245 6244 WEA /RESTORE INTERPTER 0221 2246 6249 ND RTJ, HLT //RESTORE INTERPTER 0222 2247 3777 VN, 3777 ///RESTORE CHARA //RESTORE CHARA 0223 //KBD PROCESSING ////////////////////////////////////						1		
0213 2241 1120 ADA 1 /RESTORE AC 0214 2242 0000 AC 0 /INHIBIT ZERO CLOBBERING 0215 2243 0000 AC 0 /INUSERS MEMORY SECMENT 0216 2244 0500 RTJ, HLT /BECOMES JMP TO USER PROG. 0222 0246 0000 RTJ, HLT /BECOMES JMP TO USER PROG. 0221 0246 0000 RTJ, HLT /BECOMES JMP TO USER PROG. 0222 0246 0000 RTJ, HLT /BECOMES JMP TO USER PROG. 0222 0246 0000 RTJ, HLT /BECOMES JMP TO USER PROG. 0221 0247 0251 0000 PC /IFAN /ISTONE CHAR. 0222 0240 0252 0000 PC /IFAN /ISTONE CHAR. /ISTONE CHAR. 0223 0252 0000 PC 0 /IFAN /ISTONE CHAR. /ISTONE CHAR. 0231 0252 0000 PC 0 /IFAN /ISTONE CHAR. /ISTONE CHAR. <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>								
2214 2242 0000 AC, 0 /initian 2215 2243 2000 DJR /initian 2217 2243 2000 DJR /initian 2217 2245 6244 APFA APFA /RESTORE INTERPT BUFFER 2210 2245 6244 APFA /RESTORE INTERPT BUFFER //initian 2210 2245 6247 3777 0VN, 3777 //initian //initian 2221 2247 3777 0VN, 3777 //initian //initian //initian 2222 2252 2325 00KBD, ROR I 5 /PUT TRAPPED INSTRUCTION //initian 2224 2252 2252 2325 00KBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2226 //initian //isitan //isitan //isitan //isitan 2230 2252 20252 00KBD, ROR I 5 /PUT TRAPPED INSTRUCTION //isitan 2231 2253 2435 247 Alstan //isitan //isitan 2232 2254 6323 JMP LOO /YES, DELIVER CHARACTER						T	N 1 0	/RESTORE AC
2215 2243 3006 D,R /INHIBIT_EERO_CLOBERING 2216 2244 6522 IDB /INUSERS_MEMORY_SEGMENT 2227 2245 6244 RMFA /RESTORE INTERUPT BUFFER 2228 2246 2000 RTJ, HLT /BECOMES_JMP_T0_USER_PROG. 2229 2246 2026 //INUSERS_MEMORY_SEGMENT 2220 2247 2777 DVN, 3777 2221 2247 2352 00KBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2226 //BD PROCESSING /ULNC BIT OMPUTER 2227 2251 8225 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2228 2254 8252 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2229 2252 8252 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2217 2251 8252 JMP LOO /YES.DELIVER IT TO USER 2233 2255 8435 LP, KST I /NO, FRESH CHARACTER? ////////////////////////////////////				۸C .		•		protection and the rest tasks in the
10 10 <td< td=""><td></td><td></td><td></td><td>AU,</td><td></td><td></td><td></td><td>ATNUTE TT AFRO CLOBBERING</td></td<>				AU,				ATNUTE TT AFRO CLOBBERING
227 7245 224 AWFA /RESTORE INTERRUPTOUFFER 220 0246 0000 RTJ, HLT /BECOMES JMP TO USER PROG. 221 224 252 0000 RTJ, HLT /BECOMES JMP TO USER PROG. 222 /252 0000 RTJ, HLT /BECOMES JMP TO USER PROG. 222 /252 0000 RTJ, HLT /BECOMES JMP TO USER PROG. 222 /252 0000 RTJ, HLT /BECOMES JMP TO USER PROG. 222 /252 0250 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 222 /252 0250 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 222 /252 0250 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2230 0250 0260 UPC, 0 /IF ANY 2231 0250 0260 UPC, 0 /IF ANY 2232 2254 6323 JMP LOO /YES, DELIVER IT OUSER 2236 0260 6255 JMP LOO /NO. RESURATER 2243 0261 6323 JMP LOO /NO. RESURATER <td< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>								
3220 3246 2000 RTJ, HLT /BECOMES JMP TO USER PROG. 2221 2247 3777 OVN, 3777 2223 /KBO PROCESSING //ITAD COMPUTER 224 2252 2325 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2224 2252 2325 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2226 //IS DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2226 2252 2325 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2226 //IS DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2226 2257 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2226 2257 2453 DDK I 1 /GET LEFTOVER CHAR. 2231 2255 2450 AZE /LEFTOVER CHAR. 2232 2254 6323 JMP LOO /YES, GO READ IT 2332 2257 0452 JMP LP /YES, HANG IN THERE 2340 2261 6323 JMP LO /NO, RETURN TO USER H. AC 2423 2261 6323 JMP LO /NOLEARED <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>RESTORE INTERRUPT BUFFER</td>								RESTORE INTERRUPT BUFFER
221 2247 3777 0VN, 3777 222 /KBD PROCESSING 2224 252 325 DOKBD, ROR I 5 /PUT TRAPPED INSTRUCTION 2226 /IBIT INTO COMPUTER //INC BIT 2226 252 0252 0000 UPC, 0 //IRC BIT 2231 2253 0450 AZE //IF ANY 2233 2255 0435 LP, KST I /NO, FRESH CHARACTER? 2234 0250 0250 JMP LOO /YES, OCREAD IT 2235 2257 0452 LZE /NO, FRESH CHARACTER? 2236 0250 0452 JMP LOO /YES, ADASE? 2236 2260 6255 JMP LO /NO, FRESH CHARACTER? 2242 2261 6323 JMP LO /NO, RESH CHARACTER? 2242 2261 6323 JMP LO /NO, RESH CHARACTER? 2242 // READ AND TRANSLATE A KEYBOARD CHARACTER ////////////////////////////////////				RTJ,				/BECOMES JMP TO USER PROG.
223 /KBD PROCESSING 224 252 0325 DOKBD, ROR I > 226 /I BIT INTO COMPUTER 227 251 1020 LDA I //IRCBIT 226 //IRCBIT //IF ANY 221 2253 0450 AZE //IF ANY 223 2254 6323 JMP LOO /YES. DELIVER CHARACTER? 223 2255 0435 LP, KST I /NO. FRESH CHARACTER? 233 2257 0452 LZE /NO. FRESH CHARACTER? 234 0260 6255 JMP LP /YES. GO READ IT 235 240 6262 JMP LP /YES. GO READ IT 236 2260 6255 JMP LP /YES. GO READ IT 237 70452 LZE /NO. FRESH CHARACTER 7 243 261 6323 JMP LOO /NO. RETURN TO USER H. AC 244 2262 0211 GRC, CLR ////////////////////////////////////		0247	3777	OVN,	3777			
0224 0252 0325 DOKBD, ROR I 5 /PUT TRAPED INSTRUCTION 0225 /I BIT INTO COMPUTER /I BIT INTO COMPUTER 0226 /I DA I /SET LEFTOVER CHAR, 0231 0253 0450 AZE /LEFTOVER CHARACTER? 0232 0254 6323 JMP LOO /YES. DELIVER CHARACTER? 0233 2255 0435 LP. KST I /NO. FRESH CHARACTER? 0234 0256 6262 JMP GRC /YES. GO READ IT 0235 0250 6425 LZE /NO. FRESH CHARACTER? 0236 0260 6255 JMP LP /YES. HANG IN THERE 0237 //ILL KEY STRUCK. //ILL KEY STRUCK. ///ILL KEY STRUCK. 0241 2261 6323 JMP LOO /NO. RETURN TO USER W. AC 0242 // /////YES. HANG IN THERE /////YES. HANG IN THERE /////YES. HANG IN THERE 0241 /////YES. 108 /////YES. HANG IN THERE /////YES. HANG IN THERE 0242 /////YES. /////YES. HANG IN THERE /////YES. HANG IN THERE //////Y	0222							
2225 /I BIT INTO COMPUTER 2226 /LINC BIT 2227 2251 1920 LDA I /GET LEFTOVER CHAR, 2231 2253 2450 AZE /LEFTOVER CHARACTER? 2232 2254 6323 JMP LDO /YES. DELIVER IT TO USER 2233 2255 6435 LP. KST I /NO. FRESH CHARACTER? 2234 0256 6262 JMP GRC /YES. ODELIVER IT TO USER 2235 2257 0452 LZE /NO. PAUSE? GO ALT 2235 2261 6323 JMP LP /YES. MARG IN THERE 2242 2261 6323 JMP LO /NO. RETURN TO USER W. AC 2243 2261 6323 JMP LO /NO. RETURN TO USER W. AC 2244 2262 Ø11 GRC, CLR // CALCULATE CHARACTER 2244 2262 Ø21 GRC, CLR // CALCULATE CHARACTER 2247 2263 2500 GRC, CLR // CALCULATE CHARACTER 2244 2262 Ø264 GRC, CLR // CALCULATE CHARAC								
0226 /LINC BIT 0227 C251 1020 LDA I /GET LEFTOVER CHAR, 0230 0252 0000 UPC, 0 /IF ANY 0231 0253 0450 AFE /LEFTOVER CHARACTER? 0232 0254 0523 JMP LOO /YES. DELIVER IT TO USER 0233 0255 0435 LP, KST I /NO. FRESH CHARACTER? 0234 0256 0260 0255 JMP GRC /YES. GO READ IT 0235 0257 0452 LZE /NO. FRESH CHARACTER? 0236 0260 6255 JMP LP /YES. HANG IN THERE 0237 0261 6323 JMP LOO /NO. RETURN TO USER W. AC 0241 0261 6323 JMP LOO /NO. RETURN TO USER W. AC 0244 0262 0211 GRC, CLR ////////////////////////////////////		2222	0325	DOKBD,	ROR	I	5	
0227 0251 1020 LDA I /GET LEFTOVER CHAR, 0230 0252 0000 UPC, 0 /IF ANY 0231 0252 0000 UPC, 0 /IF ANY 0232 0255 0450 AZE /LEFTOVER CHARACTER? 0233 0255 0435 LP, KST I /NO.FRESH CHARACTER? 0234 0256 6262 JMP CVES. GO READ IT 0235 0257 0452 LZE /NO.FRESH CHARACTER? 0236 0260 6255 JMP LP /YES. GO READ IT 0236 0261 6323 JMP LOO /NO.FRESH CHARACTER? 0240 0261 6323 JMP LOO /NO.RETURN TO USER W.AC 0241 0261 6323 JMP LOO /NO.RETURN TO USER W.AC 0242 / /READ AND TRANSLATE A KEYBOARD CHARACTER 0244 0264 6034 KRSA /READ A CHARACTER 0242 0264 6034 KRSA /READ A CHARACTER 0252 0272 120 ADA I /CA								
0230 0292 0000 UPC, 0 /IF ANV 0231 0252 0000 UPC, 0 /IF ANV 0231 0253 0450 AZE /LEFTOVER CHARACTER? 0232 0255 0435 LP, KST I /NO, FRESH CHARACTER? 0234 0256 6262 JMP GRC /YES. GO READ IT 0235 0257 0452 LZE /NO, PAUSE? YO, PAUSE? 0237 0261 6323 JMP LP /YES. HANG IN THERE 0237 2261 6323 JMP LOO /NO, RETURN TO USER W. AC 0241 0261 6323 JMP LOO /NO, RETURN TO USER W. AC 0242 / / /READ AND TRANSLATE A KEYBOARD CHARACTER 0244 0242 // // // // // 0244 2262 0204 GRC, CLR // // 0242 0264 6234 KRSA /READ A CHARACTER // 0250 0264 6234 KRSA <td></td> <td>0 0 E 1</td> <td>1000</td> <td></td> <td></td> <td>T</td> <td></td> <td></td>		0 0 E 1	1000			T		
2231 2253 2450 AZE /LEFTOVER CHARACTER? 2232 2254 6323 JMP L00 /YES. DELIVER IT TO USER 2233 2255 6435 LP, KST I /NO. FRESH CHARACTER? 2234 0256 6262 JMP GRC /YES. GO READ IT 2235 2267 6452 LZE /NO. FRESH CHARACTER? 2236 2260 6255 JMP LP /YES. HANG IN THERE 2237 2261 6323 JMP LO /NO. FRESH CHARACTER? 2243 2261 6323 JMP LO /NO. RETURN TO USER W. AC 2243 2261 6323 JMP LO /NO. RETURN TO USER W. AC 2243 2261 6323 JMP LO /NO. RETURN TO USER W. AC 2243 2264 2262 Ø211 GRC, CLR ////////////////////////////////////		• • •		UP C.		1		
2232 2254 6323 JMP LOO /YES. DELIVER IT TO USER 2233 2255 0435 LP, KST I /NO. FRESH CHARACTER? 2234 0256 6262 JMP GRC /YES. GO READ IT 2235 0257 0452 LZE /NO. PAUSE? 2236 0260 6255 JMP LP /YES. HANG IN THERE 2237 0261 6323 JMP LOO /NO. RETURN TO USER W. AC 0243 0261 6323 JMP LOO /NO. RETURN TO USER W. AC 0243 0261 6323 JMP LOO /NO. RETURN TO USER W. AC 0241 0262 0211 GRC. CLR 0244 0262 0211 GRC. CLR 0244 0264 0234 KRSA /READ A CHARACTER 0247 0265 1120 ADA I /CALCULATE CHARACTER CON- 0251 0267 0391 ROR 1 /HALF WORDS 0252 0273 0401 TAB ////////////////////////////////////				01 07	-			-
2233 2255 0435 LP, KST I /NO, FRESH CHARACTER? 2234 2256 6262 JMP GRC /YES, GO READ IT 2235 2257 0452 LZE /NO, PAUSE? 2236 2260 6255 JMP LP /YES, HANG IN THERE 0237 0242 0261 6323 JMP LOO /NO, RETURN TO USER W. AC 0243 0261 6323 JMP LOO /NO, RETURN TO USER W. AC 0244 0262 0011 GRC, CLR ///CLEARED ////////////////////////////////////							L00	
0234 0256 6262 JMP GRC /YES, GO READ IT 0235 0257 0452 LZE /NO, PAUSE? 0237 JMP LP /YES, HANG IN THERE 0237 JMP LO /NO, RETURN TO USER W. AC 0243 0261 6323 JMP LO /NO, RETURN TO USER W. AC 0241 ////////////////////////////////////			Ø435	LP,	KST	I		
2236 2260 6255 JMP LP /YES. HANG IN THERE 0237 JMP LO /NO. RETURN TO USER W. AC 0242 JMP LOO /NO. RETURN TO USER W. AC 0242 / /READ AND TRANSLATE A KEYBOARD CHARACTER 0244 0264 0264 0265 108 0245 0265 120 ADA /KEAD AND TRANSLATE A KEYBOARD CHARACTER 0244 0266 6034 KRSA /READ A CHARACTER 0245 0265 1120 ADA /CALCULATE CHARACTER CON- 0250 0266 7602 -175 /VERSION TABLE INDEX IN 0251 0267 0331 ROR /HALF WORDS 0252 2272 1560 BCL I 0255 2273 0401 TAB 0256 2274 4276 STC .+2 /HALF-WORD PICKUP ADDRESS 0257 0275 1380 LOH /IS IT AN UNDEFINABLE? 0262 2276 0200 Ø /IS IT AN UNDEFINABLE? 0261 2277 1420 SHD	Ø234	0256	6262		JMP		GRC	
0237 0261 6323 JMP LOO /NO. RETURN TO USER W. AC 0241 //0241 //00000000000000000000000000000000000	0235		Ø452					
0240 0261 6323 JMP LOO /NO, RETURN TO USER W. AC 0241 //CLEARED 0242 / 0243 /READ AND TRANSLATE A KEYBOARD CHARACTER 0244 0262 0211 GRC, CLR 0245 7263 0500 IOB 0246 0264 6034 KRSA /READ A CHARACTER 0247 0265 1120 ADA I /CALCULATE CHARACTER CON- 0250 0266 7602 -175 /VERSION TABLE INDEX IN 0251 3267 0301 ROR 1 /HALF WORDS 0252 0271 3700 3702		226Ø	6255		JMP	~	LP	
0241 /CLEARED 0242 /READ AND TRANSLATE A KEYBOARD CHARACTER 0243 (READ AND TRANSLATE A KEYBOARD CHARACTER) 0244 0262 0211 0245 0263 0500 0246 0264 6034 KRSA 0247 0265 1120 ADA I 0250 0266 7602 -175 /VERSION TABLE INDEX IN 0251 0267 0301 ROR 1 /HALF WORDS 0252 0273 1560 BCL I 0253 0271 1700 3702 0254 0272 1120 ADA I /ADD BASE ADDRESS OF TABLE 0255 0273 0401 TAB ////////////////////////////////////		2244	(707					
0242 //READ AND TRANSLATE A KEYBOARD CHARACTER 0244 0262 0011 GRC, CLR 0245 0263 0264 6034 KRSA 0246 0264 6034 KRSA /READ A CHARACTER 0247 0265 1120 ADA I /CALCULATE CHARACTER CON- 0250 0266 7602 -175 /VERSION TABLE INDEX IN 0251 0267 0301 ROR 1 /HALF WORDS 0252 0270 1560 BCL I 0253 0271 3700 0253 0271 1700 3702 0254 0272 1120 ADA I /ADD BASE ADDRESS OF TABLE 0255 0273 0401 TAB 0255 0273 0401 TAB 0256 0274 4276 STC .+2 /HALF-WORD PICKUP ADDRESS 0261 0277 1420 SHD I /IS IT AN UNDEFINABLE? 0262 0275 1300 LDH /GET THE LINC CHARACTER, 0264 0302 1120 ADA I /IS IT UPPER CASE? 02		0201	0323		JMP		LUU	
7READ AND TRANSLATE A KEYBOARD CHARACTER 0243 7READ AND TRANSLATE A KEYBOARD CHARACTER 0244 0262 0211 GRC, CLR 0245 7263 0500 10B 0246 0264 6034 KRSA /READ A CHARACTER 0247 0265 1120 ADA I /CALCULATE CHARACTER CON- 0250 0266 7602 -175 /VERSION TABLE INDEX IN 0251 3267 0301 ROR 1 /HALF WORDS 0252 2272 1560 BCL I 0253 0211 3702 0254 2272 1120 ADA I /ADD BASE ADDRESS OF TABLE 0255 0273 0401 TAB //eet THE LINC CHARACTER, 0256 2274 4276 STC .+2 /HALF-WORD PICKUP ADDRESS 0257 0275 1300 LDH /GET THE LINC CHARACTER, 0262 2327 1420 SHD I /IS IT AN UNDEFINABLE? 0262 2332 7620 7620 //ex 0264 0302 1120 ADA I /IS IT UPPER CAS				1				
0244 0262 0011 GRC, CLR 0245 0263 0500 IOB 0246 0264 6034 KRSA /READ A CHARACTER 0247 0265 1120 ADA I /CALCULATE CHARACTER CON- 0250 0267 0301 ROR 1 /HALF WORDS 0251 0267 0301 ROR 1 /HALF WORDS 0252 0270 1560 BCL I 0253 0271 3700 3702 0254 0272 1120 ADA I /ADD BASE ADDRESS OF TABLE 0255 0273 0401 TAB /ADD BASE ADDRESS OF TABLE 0255 0273 0401 TAB /ADD BASE ADDRESS OF TABLE 0256 0274 4276 STC .+2 /HALF-WORD PICKUP ADDRESS 0257 0275 1300 LDH /GET THE LINC CHARACTER, 0262 0276 0200 Ø // 0261 2277 1420 SHD I /IS IT AN UNDEFINABLE? 0262 0302 762					ND TR		ΔΤΕ Δ ΚΙ	EYBOARD CHARACTER
0245 0263 0500 IOB 0246 0264 6034 KRSA /READ A CHARACTER 0247 0265 1120 ADA I /CALCULATE CHARACTER CON- 0250 0266 7602 -175 /VERSION TABLE INDEX IN 0251 0267 0301 ROR 1 /HALF WORDS 0252 2272 1560 BCL I 0253 0271 3700 3702 0254 2272 1120 ADA I /ADD BASE ADDRESS OF TABLE 0255 0273 0401 TAB ////////////////////////////////////		0262	0011					
0246 0264 6034 KRSA /READ A CHARACTER 0247 0265 1120 ADA I /CALCULATE CHARACTER CON- 0250 0266 7602 -175 /VERSION TABLE INDEX IN 0251 0267 0301 ROR 1 /HALF WORDS 0252 0270 1560 BCL I 0253 0271 3700 3707 0254 2272 1120 ADA I /ADD BASE ADDRESS OF TABLE 0255 0273 0401 TAB 0256 2274 4276 STC /HALF-WORD PICKUP ADDRESS 0256 2274 4276 STC /HALF-WORD PICKUP ADDRESS 0262 2275 1300 LDH /GET THE LINC CHARACTER, 0262 2277 1420 SHD I /IS IT AN UNDEFINABLE? 0262 2302 7600 0264 0302 1120 ADA I /IS IT UPPER CASE? 0265 <			-					
0250 0266 7602 -175 /VERSION TABLE INDEX IN 0251 0267 0301 ROR 1 /HALF WORDS 0252 0270 1560 BCL I 0253 0271 3700 3702 0254 0272 1120 ADA I /ADD BASE ADDRESS OF TABLE 0255 0273 0401 TAB //ADD BASE ADDRESS OF TABLE 0256 0274 4276 STC .+2 /HALF-WORD PICKUP ADDRESS 0257 0275 1300 LDH /GET THE LINC CHARACTER, 0262 0276 0200 0 0 0261 0277 1420 SHD I /IS IT AN UNDEFINABLE? 0262 0302 7620 7620 0 0263 0301 6337 JMP RETX /YES. ECHO UPARROW 0264 0302 1120 ADA I /IS IT UPPER CASE? 0265 0303 7722 +55 0 0266 0304 0451 APO 0 0267 0305 6317	0246	0264	6034		KRSA			
Ø251 3267 Ø301 ROR 1 /HALF WORDS Ø252 2270 1560 BCL I Ø253 0271 3700 3702 Ø254 0272 1120 ADA I /ADD BASE ADDRESS OF TABLE Ø255 0273 Ø401 TAB ////////////////////////////////////	0247	0265	112Ø			I		
0252 2270 1560 BCL I 0253 0271 3700 3702 0254 2272 1120 ADA I / ADD BASE ADDRESS OF TABLE 0255 0273 0401 TAB 0256 2274 4276 STC .+2 / HALF-WORD PICKUP ADDRESS 0257 0275 1300 LDH / GET THE LINC CHARACTER, 0262 0276 0200 0 0261 2277 1420 SHD I / IS IT AN UNDEFINABLE? 0262 0302 7600 7600 0263 2301 6337 JMP RETX 0264 0302 1120 ADA I / IS IT UPPER CASE? 0265 0303 7722 +55 0 0266 0304 0451 APO 0 0267 0305 6317 JMP LOW /NO, 0270 / / ITS AN UPPER CASE CHARACTER 0 0271 // ITS AN UPPER CASE CHARACTER 0 4 ADA I / MAKE IT LINC CODE								
Ø253 Ø271 3700 3702 Ø254 Ø272 1120 ADA I / ADD BASE ADDRESS OF TABLE Ø255 ©273 Ø401 TAB	-						1	/HALF WORDS
Ø254 $\partial 272$ 1120 ADA I /ADD BASE ADDRESS OF TABLE Ø255 $\partial 273$ $\partial 401$ TAB TAB TAB Ø256 $\partial 274$ 4276 STC .+2 /HALF-WORD PICKUP ADDRESS Ø257 $\partial 275$ 1300 LDH /GET THE LINC CHARACTER, Ø262 $\partial 276$ $\partial 200$ ∂ ∂ ∂ ∂ ∂ Ø261 $\partial 277$ 1420 SHD I /IS IT AN UNDEFINABLE? Ø262 $\partial 302$ 7620 7600 ∂ ∂ ∂ ∂ Ø263 $\partial 301$ 6337 JMP RETX /YES. ECHO UPARROW Ø264 $\partial 3d2$ 1120 ADA I /IS IT UPPER CASE? Ø265 0304 0451 APO 0267 0305 6317 JMP LOW /NO. Ø270 / / ITS ADA I /						1		
0255 0273 0401 TAB 0256 2274 4276 STC .+2 /HALF-WORD PICKUP ADDRESS 0257 0275 1300 LDH /GET THE LINC CHARACTER, 0262 0276 000 0 0261 0277 1420 SHD I /IS IT AN UNDEFINABLE? 0262 0302 7600 7600 0 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 0267 0305 6317 JMP LOW 0270 / / ITS AN UPPER CASE CHARACTER 0271 /ITS AN UPPER CASE CHARACTER 0306 1120						Ŧ		AND PASE ADDRESS OF TABLE
0256 0274 4276 STC .+2 /HALF-WORD PICKUP ADDRESS 0257 0275 1300 LDH /GET THE LINC CHARACTER, 0260 0276 0000 0 0261 0277 1420 SHD I /IS IT AN UNDEFINABLE? 0262 0302 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 0267 0305 6317 JMP LOW /NO, 0270 / //ITS AN UPPER CASE CHARACTER //IS IT LINC CODE 0271 //ITS AN UPPER CASE CHARACTER 0306 1120 ADA I //MAKE IT LINC CODE	-					1		AND BASE ADDITEDS OF TABLE
0257 0275 1300 LDH /GET THE LINC CHARACTER, 0260 0276 000 0 0 0261 0277 1420 SHD I /IS IT AN UNDEFINABLE? 0262 0302 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 0267 0305 6317 JMP LOW /NO. 0270 / //ITS AN UPPER CASE CHARACTER 0271 //ITS AN UPPER CASE CHARACTER 0272 0306 1120 ADA I /MAKE IT LINC CODE							+2	ZHALE-WORD PICKUP ADDRESS
0260 0276 000 0 0261 0277 1420 SHD I / IS IT AN UNDEFINABLE? 0262 2302 7600 7600 7600 0263 2301 6337 JMP RETX / YES. ECHO UPARROW 0264 0302 1120 ADA I / IS IT UPPER CASE? 0265 0303 7722 ~55								
0261 3277 1420 SHD I /IS IT AN UNDEFINABLE? 0262 2332 7630 7600 0263 2301 6337 JMP RETX /YES. ECHO UPARROW 0264 0302 1120 ADA I /IS IT UPPER CASE? 0265 0303 7722 +55 0266 0304 0451 APO 0267 0305 6317 JMP LOW /NO, 0270 / //ITS AN UPPER CASE CHARACTER //ITS AN UPPER CASE CHARACTER 0272 0306 1120 ADA I //MAKE IT LINC CODE								
0262 2332 7620 7600 0263 2301 6337 JMP RETX /YES. ECHO UPARROW 0264 0302 1120 ADA I /IS IT UPPER CASE? 0265 0303 7722 +55					SHD	I		/IS IT AN UNDEFINABLE?
0264 0302 1120 ADA I /IS IT UPPER CASE? 0265 0303 7722 +55					7600			
0265 0303 7722 +55 0266 0304 0451 APO 0267 0305 6317 JMP LOW /NO, 0270 / 0271 /ITS AN UPPER CASE CHARACTER 0271 /ITS AN UPPER CASE CHARACTER 0272 0306 1120 ADA I /MAKE IT LINC CODE	0263	2301	6337		JMP		RETX	
0266 0304 0451 APO 0267 0305 6317 JMP LOW /NO, 0270 / 0271 /ITS AN UPPER CASE CHARACTER 0272 0306 1120 ADA I /MAKE IT LINC CODE						I		/IS IT UPPER CASE?
0267 0305 6317 JMP LOW /NO, 0270 / 0271 /ITS AN UPPER CASE CHARACTER 0272 0306 1120 ADA I /MAKE IT LINC CODE		-						
0270 / 0271 /ITS AN UPPER CASE CHARACTER 0272 0306 1120 ADA I /MAKE IT LINC CODE							1.00	(NO
0271 /ITS AN UPPER CASE CHARACTER 0272 0306 1120 ADA I /MAKE IT LINC CODE		0300	001/	,	JMP		LUW	Z NU +
0272 0306 1120 ADA I /MAKE IT LINC CODE						~ 6	ASE CHAR	ACTER
		UZDA	1120	VIIS AN				
						*		

ŧ

0070 0077			/TRAP E	ENTRY A	ND	MACHINE	STATE SAVING
0100			* 141				(0.1.V.E. 1.0.
0101	0141	4242		STC	Ŧ	AC	/SAVE AC /SAVE LINK
0102	Ø142 Ø143	0261 4225		ROL STC	I	1 LC	JAVE LINK
Ø1Ø3 Ø1Ø4	0144	0005		QAC			/GET H O 11 Q REG BITS
Ø1Ø5 Ø1Ø6	Ø145 Ø146	Ø241 Ø455		ROL QLZ		1	/GET L O BIT
0107	0147	2136		ADD		ONE	(0.) VE 17
0110	Ø150	4232 2136		STC	NE	Q	/SAVE IT
Ø111 Ø112	Ø151 Ø152	2130 Ø454		ADD O FLO			/GET OVERFLOW BIT IF ON
0112	Ø153	0011		CLR			
Ø114	Ø154	4222		STC		OVL	/SAVE IT,Ø=OFF, 1=ON
0115	Ø155	0057·		SET		17	/SAVE LOCATION Ø
Ø116	Ø156	0000		0			
0117			/	OTION			
Ø12Ø	0157	0002	/INSTRU	PDP	IDE	INTIFICAT	ITON.
Ø121 Ø122	0107	0002		PMODE			
Ø123	4160	6234	FETCH,	RIB			/READ INTERRUPT BUFFER
Ø124	4161	0335		AND		M70	/GET INSTRUCTION FIELD
0125	4162	1371		TAD		CDFX	/MAKE IT INTO A CDF N
0126	4163	3364		DCA		.+1	
0127	4164	7402		7402			/BECOMES CDF N /SET AC TO -1
Ø13Ø Ø131	4165 4166	724Ø 1140		STA TAD		140	COMPUTE ADDRESS OF
Ø131 Ø132	4167	3334		DCA		PC	/TRAPPED INSTRUCTION
Ø133	4170	1734			I	PC	OBTAIN INSTRUCTION
Ø134							/CAUSING TRAP
0135	4171	6201	CDFX,	CDF			/SET DATA FIELD TO Ø
0136	4172	6141		LINC			
0137	0.177	4.4.4.3		LMODE SAE	I		/IS IT KBD?
Ø14Ø Ø141	Ø173 Ø174	1460 0515		KBD	Ŧ		715 11 (60)
0142	0175	0467		SKP			/N0
0143	Ø176	6250		JMP		DOKBD	/YES.
0144	Ø177	1460			I		/IS IT KBD I?
0145	0200	0535		KBD	I		(1)0
0146	0201	Ø467 625Ø		SKP JMP		роквр	/NO, /YES,
Ø147 Ø150	Ø2Ø2 Ø2Ø3	1460			I	DUNBD	/OPR I 14?
0151	0204	0534		OPR	Î	14	
0152	Ø2Ø5	0467		SKP			/N0
Ø153	0206	6212		JMP	-	0P14	/YES
0154	0207			SAE	1	1 4	/OPR 14?
Ø155	Ø21Ø Ø211	0514 6130	OTHERS			14 ERR	/PUT JMP TO
Ø156 Ø157	UZII	0100	UINCIUI	, ,,,,,			/CHECKS FOR OTHER
Ø16Ø							/INSTRUCTIONS HERE,
0161	0212	1000	OP14,	LDA			/OUTPUT, GET CHARACTER
0162	0213	Ø242		AC			
0163	Ø214	0520		IOB			
0164	0215	6046 0500		TLSA			/OUTPUT IT
Ø165 Ø166	Ø216 Ø217	0500 6041		IOB TSFA			/CHECK IF DONE
Ø166	0210	6216		JMP		, -2	
0170	~ ~ ~ ~ ~		1			-	
0171			/RETURN		ER	PROGRAM	
0172	O 221	1020	RET,	-	I		VEXIT TRAP PROC
0173	0222	0000	OVL,	Ø		/3777 C/	
0174	0223	2247	ADD	OVN		13/// 64	

0 0				/ /COPYRI /MAYNAR / /D, LAN /REVISE /REVISE / /OPERAT	GHT 1969 D, MASS. GBEIN, 27 D 21 JULN D 8 AUGUS ION DEFIN -8 MODE 1 32 34	, DIGITAL 7 MAY 196 7 1909 51 1969 9 1110NS F	FOR LINC MODE ASSEMBLY				
001 002 002 002 002 002 002 002 002	9 2 3 4 5 5			TSFA=6041 TLSA=6046 RMFA=6244 / /SOME LINC-8 DEFINITIONS OPR=500 KBD=515 /							
002 003 003 003 003 003 003 003) L 2 3 4	0020 0021 0022 0023 0024	6463 0643 0602 0006 6020	*22	JMP LDF LIF DJR JMP	ST20 3 2 20	/BECOMES JMP INIT /GO TO 20 IN SEGMENT 2 /(04020)				
003 003 004 004 004 004 004 004	5 7 1 2 3	0025 0026 0027 0030 0031	6466 2643 2622 2226 2226 0220	/ /400 RE RST400,		ST 400 3 2	/BECOMES JMP INIT				
004 004 004 004 005	5 6 7	ØØ32	6400 6400	/ /start	JMP	400	/GO TO 400 IN SEGMENT 2 /(04400)				
ØØ5: ØØ5: ØØ5: ØØ5: ØØ5:	L 2 3 4 5 5	0033 0034 0035 0036 0037 0040	0642 0/00 7400 0643 0602	GUIDE,	JMP LDF RDC 74Ø0 LDF LIF	INIT 2 3 2	/READ GUIDE START BLOCK /INTO QUARTER 3,SEGMENT 2				
ØØ5 ØØ6 ØØ6 ØØ6	0 1 2	0041	7400	/ /ERROR #130		1400	/START GUIDE				
ØØ6 ØØ6 ØØ6 ØØ6	4 5 5	Ø130 Ø131	0000 6221	ERR,	HLT JMP	REŤ	/UNIDENTIFIED INSTRUCTION /IN AC AT HLT, RETURNS TO /USER W AC CLEARED WHEN /CONTINUE PRESSED,				
007 007 007 007	ð 1 2 3	Ø134	0000 0070	*134 PC;	ONSTANTS Ø		ADDRESS OF TRAPPED				
007 007		Ø135 Ø136	0070 0001	M7Ø, ONE,	7Ø 1						

,

 \bigcirc

()0 .