

digital

**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

CONTENTS

	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	

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 adaptations are provided in this document, along with enough information on the internal operation of the program to permit users to easily implement adaptations 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-SI1B-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.

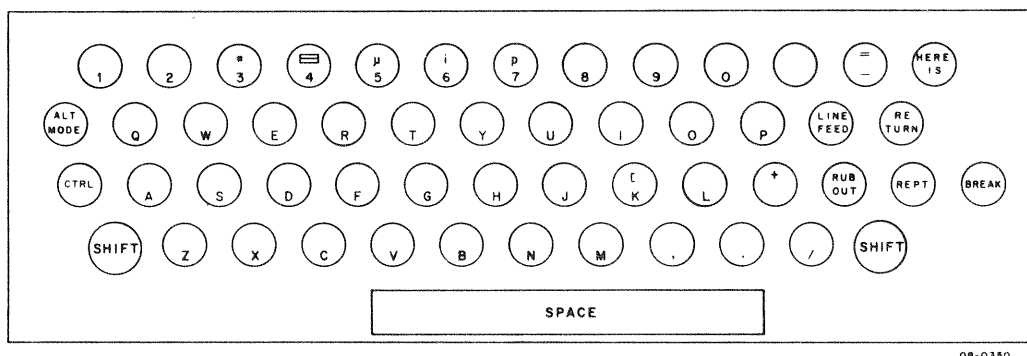


Figure 1. Location of LINC keys on the Teletype keyboard

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:

!, ", TAB, (,), *, ?, @, I, ↑, ;, <, >.

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 relocater routine at symbolic location SETUP. This routine relocates the trap processor from the current instruction field into memory segment 0. Notice that the relocater 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 save 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 OP14. 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.)

```

/
/SUBR TO RUN XY12 CONTROL FROM LINC
/MODE USING TRAPPED OPRS.
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
           PLCF
PLGO,      HLT            /PLPU,PLPR, ETC
           PLSF
           JMP .-1
           LINC
           LMODE
           JMP RET        /RETURN TO USER
/
/INSTRUCTION NOT RECOGNIZED
PLERR,     LDA
           PLINST
           JMP ERR        /DO ERROR STOP
           PDP
           PMODE
PLTAB,     PLPD            /PEN DOWN. OPR 5
           PLPU            /PEN UP
           PLDU            /DRUM DOWN
           PLDD            /DRUM UP
           PLPR            /PEN RIGHT
           PLPL            /PEN LEFT .OPR 12
           LINC
           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 SEGMENT 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	INIT	/INITIALIZE TRAP PROCESSOR
	LDF	2	
	LDA	I	/PUT RCG 7300 IN
	RCG		/4016 AND 4017
	STA		
	2016		
	LDA	I	
	7300		
	STA		
	2017		
	LDF	3	
	LIF	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:

L00,	STC	AC	/STORE IT FOR TRANSFER
	STC	UPC	/CLEAR LEFT OVER BUFFER
	IOB		
	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.

```

0000      *20
0001      /LINC-8 SIMULATOR TRAP PROCESSOR, DEC-12-S11B.
0002      /
0003      /COPYRIGHT 1969, DIGITAL EQUIPMENT CORP.
0004      /MAYNARD, MASS.
0005      /
0006      /D, LANGBEIN, 27 MAY 1969
0007      /REVISED 21 JULY 1969
0010      /REVISED 8 AUGUST 1969
0011      /
0012      /OPERATION DEFINITIONS FOR LINC MODE ASSEMBLY
0013      /OF PDP-8 MODE IOT INSTRUCTIONS
0014      KCCA=6032
0015      KRSA=6034
0016      KRBA=6036
0017      TSFA=6041
0020      TLSA=6046
0021      RMFA=6244
0022      /
0023      /SOME LINC-8 DEFINITIONS
0024      OPR=500
0025      KBD=515
0026      /
0027      *20
0030      0020 6463      JMP      ST20      /BECOMES JMP INIT
0031      0021 0643      LDF      3
0032      0022 0602      LIF      2
0033      0023 0006      DJR
0034      0024 6020      JMP      20      /GO TO 20 IN SEGMENT 2
0035                               / (04020)
0036      /
0037      /400 RESTART
0040      0025 6466      RST400, JMP      ST400      /BECOMES JMP INIT
0041      0026 0643      LDF      3
0042      0027 0602      LIF      2
0043      0030 0006      DJR
0044      0031 0000      HLT
0045      0032 6400      JMP      400      /GO TO 400 IN SEGMENT 2
0046                               / (04400)
0047      /
0050      /START GUIDE
0051      0033 6356      GUIDE, JMP      INIT
0052      0034 0642      LDF      2
0053      0035 0700      RDC
0054      0036 7400      7400      /READ GUIDE START BLOCK
0055      0037 0643      LDF      3      /INTO QUARTER 3, SEGMENT 2
0056      0040 0602      LIF      2
0057      0041 7400      JMP      1400      /START GUIDE
0060      /
0061      /ERROR STOP
0062      *130
0063      0130 0000      ERR,      HLT
0064      0131 6221      JMP      RET      /UNIDENTIFIED INSTRUCTION
0065                               /IN AC AT HLT. RETURNS TO
0066                               /USER W AC CLEARED WHEN
0067                               /CONTINUE PRESSED.
0068      /
0069      /MISC CONSTANTS
0070      *134
0071      0134 0000      PC,      0      /ADDRESS OF TRAPPED
0072                               /INSTRUCTION
0073      0135 0070      M70,      70
0074      0136 0001      ONE,      1
0075

```

```

0076 /
0077 /TRAP ENTRY AND MACHINE STATE SAVING
0100 *141
0101 0141 4242 STC AC /SAVE AC
0102 0142 0261 ROL I 1 /SAVE LINK
0103 0143 4225 STC LC
0104 0144 0005 QAC /GET H 0 11 Q REG BITS
0105 0145 0241 ROL 1
0106 0146 0455 QLZ /GET L 0 BIT
0107 0147 2136 ADD ONE
0110 0150 4232 STC Q /SAVE IT
0111 0151 2136 ADD ONE
0112 0152 0454 FLO /GET OVERFLOW BIT IF ON
0113 0153 0011 CLR
0114 0154 4222 STC OVL /SAVE IT, 0=OFF, 1=ON
0115 0155 0057 SET 17 /SAVE LOCATION 0
0116 0156 0000 0
0117 /
0120 /INSTRUCTION IDENTIFICATION.
0121 0157 0002 PDP
0122 PMODE
0123 4160 6234 FETCH, RIB /READ INTERRUPT BUFFER
0124 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
0130 4165 7240 STA /SET AC TO -1
0131 4166 1140 TAD 140 /COMPUTE ADDRESS OF
0132 4167 3334 DCA PC /TRAPPED INSTRUCTION
0133 4170 1734 TAD I PC /OBTAIN INSTRUCTION
0134 /CAUSING TRAP
0135 4171 6201 CDFX, CDF /SET DATA FIELD TO 0
0136 4172 6141 LINC
0137 LMODE
0140 0173 1460 SAE I /IS IT KBD?
0141 0174 0515 KBD
0142 0175 0467 SKP /NO
0143 0176 6250 JMP DOKBD /YES.
0144 0177 1460 SAE I /IS IT KBD I?
0145 0200 0535 KBD I
0146 0201 0467 SKP /NO.
0147 0202 6250 JMP DOKBD /YES.
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 /CHECKS FOR OTHER
0160 /INSTRUCTIONS HERE.
0161 0212 1000 OP14, LDA /OUTPUT, GET CHARACTER
0162 0213 0242 AC
0163 0214 0500 IOB
0164 0215 6046 TLSA /OUTPUT IT
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 PROCESSOR
0173 0222 0000 OVL, 0
0174 0223 2247 ADD OVN /3777 CAUSES OV

```

0175	0224	1020	LOA	I	/GET LINK
0176	0225	0000	LC,	0	
0177	0226	0321	ROR	I 1	/RESTORE LINK
0200	0227	0040	SET	0	/RESTORE LOCATION 0
0201	0230	0017	17		
0202	0231	1020	LOA	I	/RESTORE Q REGISTER
0203	0232	0000	Q,	0	
0204	0233	0314	ROR	14	/LOADS Q REGISTER
0205					/FROM AC
0206	0234	1000	LOA		/OBTAIN ADDRESS FOR RETURN
0207	0235	0140	140		/TO USER PROGRAM
0210	0236	1620	BSE	I	
0211	0237	6000	JMP		
0212	0240	4246	STC	RTJ	
0213	0241	1120	ADA	I	/RESTORE AC
0214	0242	0000	AC,	0	
0215	0243	0006	DJR		/INHIBIT ZERO CLOBBERING
0216	0244	0500	IOB		/IN USERS MEMORY SEGMENT
0217	0245	6244	RMFA		/RESTORE INTERRUPT BUFFER
0220	0246	0000	RTJ,	HLT	/BECOMES JMP TO USER PROG.
0221	0247	3777	OVN,	3777	
0222			/		
0223			/KBD PROCESSING		
0224	0250	0325	DOKBD,	ROR I 5	/PUT TRAPPED INSTRUCTION
0225					/I BIT INTO COMPUTER
0226					/LINC BIT
0227	0251	1020	LOA	I	/GET LEFTOVER CHAR,
0230	0252	0000	UPC,	0	/IF ANY
0231	0253	0450	AZE		/LEFTOVER CHARACTER?
0232	0254	6323	JMP	LOO	/YES, DELIVER IT TO USER
0233	0255	0435	LP,	KST I	/NO, FRESH CHARACTER?
0234	0256	6262	JMP	GRC	/YES, GO READ IT
0235	0257	0452	LZE		/NO, PAUSE?
0236	0260	6255	JMP	LP	/YES, HANG IN THERE
0237					/TILL KEY STRUCK.
0240	0261	6323	JMP	LOO	/NO, RETURN TO USER W. AC
0241					/CLEARED
0242			/		
0243			/READ AND TRANSLATE A KEYBOARD CHARACTER		
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	0266	7602	-175		/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		
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	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	APD		
0267	0305	6317	JMP	LOW	/NO,
0270			/		
0271			/ITS AN UPPER CASE CHARACTER		
0272	0306	1120	ADA	I	/MAKE IT LINC CODE
0273	0307	0011	11		

0274	0310	0451	APD		/DELETE MINUS ZEROES
0275	0311	0011	CLR		
0276	0312	4252	STC	UPC	/SAVE IT FOR USE AFTER
0277					/CASE CHARACTER IS OUTPUT
0300	0313	1020	LDA	I	/GET CASE CHARACTER
0301	0314	0023	23		
0302	0315	4242	STC	AC	/STORE FOR TRANSFER
0303	0316	6221	JMP	RET	/EXIT
0304			/		
0305			/LOWER CASE CHARACTERS		
0306	0317	1120	LOW,	ADA I	/MAKE IT LINC CODE
0307	0320	0055		55	
0310	0321	0451	APD		
0311	0322	0011	CLR		
0312	0323	4242	LOO,	STC AC	/STORE IT FOR TRANSFER
0313	0324	4252		STC UPC	/CLEAR LEFT OVER BUFFER
0314	0325	0500		IOB	
0315	0326	6036		KRBA	/READ CHAR AND
0316					/CLEAR KEYBOARD FLAG
0317	0327	6350	LOO1,	JMP PRINT	/ECHO
0320	0330	1460		SAE I	/C, R.?
0321	0331	0215		215	
0322	0332	6221		JMP RET	/NO, EXIT
0323	0333	1020		LDA I	/YES, THROW IN
0324	0334	0212		212	/A FREE LINE FEED
0325	0335	6350		JMP PRINT	
0326	0336	6221		JMP RET	
0327			/		
0330			/ILLEGAL CHARACTER		
0331	0337	1020	RETX,	LDA I	/PRINT UP ARROW
0332	0340	0336		336	
0333	0341	6350		JMP PRINT	
0334	0342	0500		IOB	/ECHO OFFENDING
0335	0343	6036		KRBA	/CHARACTER.
0336	0344	6350		JMP PRINT	
0337	0345	0017		COM	
0340	0346	0261		ROL I 1	/SET LINK BIT
0341					/TO KEYBOARD "PAUSE" VALUE
0342	0347	6255		JMP LP	
0343			/		
0344			/PRINT A CHARACTER		
0345			/ENTER W ASCII CHAR IN AC		
0346	0350	0002	PRINT,	PDP	
0347				Pmode	
0350	4351	6046		TLS	
0351	4352	6041		TSF	
0352	4353	5352		JMP	,-1
0353	4354	6141		LINC	
0354				Lmode	
0355	0355	6000		JMP 0	
0356			/INITIALIZE THE TRAP PROCESSOR		
0357	0356	0011	INIT,	CLR	
0360	0357	4252		STC UPC	/CLEAR LEFT OVER CHARACTER
0361	0360	0001		AXO	/BOILER PLATING FOR GUIDE
0362					/RESTART
0363	0361	0024		SFA	/GET SPEC, FNS. REG.
0364	0362	1620		BSE I	/SET TRAP ENABLE BIT
0365	0363	1000		1000	
0366	0364	0004		ESF	/READ IT BACK OUT
0367	0365	6000	ENDX,	JMP 0	/RETURN
0370			*400		
0371	2400	6025		JMP RST400	/GO TO START 400
0372					/CONTROL ROUTINE

```

0373 /
0374 /
0375 0401 2323 TAB, 2323 /ALTMODE-CASE/ALTMODE-CASE
0376 0402 1376 1376 /RUBOUT-DEL/LEADER=NULL
0377 0403 7676 7676 /NOT USED
0400 0404 7676 7676 /N, U,
0401 0405 7676 7676 /N, U,
0402 0406 7676 7676 /BELL/N, U,
0403 0407 7656 7656 /TAB/L, F, -META
0404 0410 7676 7676 /N, U,
0405 0411 1276 1276 /CR-EOL/N, U,
0406 0412 7676 7676 /N, U,
0407 0413 7676 7676 /N, U,
0410 0414 7676 7676 /N, U,
0411 0415 7676 7676 /N, U,
0412 0416 7676 7676 /N, U,
0413 0417 7676 7676 /N, U,
0414 0420 7676 7676 /N, U,
0415 0421 2376 2376 /ALTMODE-CASE/N, U,
0416 0422 7614 7614 /N,U,/SPACE=SPACE
0417 0423 7676 7676 /EXC,PT/DBL,QUOTES
0420 0424 2265 2265 /NUMBER SIGN/DOLLAR SIGN-
0421 0425 6215 6215 /PERCENT-P,C/AMPERSAND-AMPERSAND
0422 0426 1676 1676 /APOSTROPHE=AP,/OP PAREN
0423 0427 7676 7676 /CL PAREM/*
0424 0430 2063 2063 /PLUS-*/COMMA-,
0425 0431 1764 1764 /MINUS--/PERIOD-,
0426 0432 2100 2100 /SLASH-SLASH/ZERO-0
0427 0433 0102 0102 /1/2
0430 0434 0304 0304 /3/4
0431 0435 0506 0506 /5/6
0432 0436 0710 0710 /7/8
0433 0437 1176 1176 /9/N,U,
0434 0440 7676 7676 /SEMICOLON/<
0435 0441 6176 6176 /EQUAL SIGN-=/>
0436 0442 7676 7676 /?/AT SIGN
0437 0443 2425 2425 /A/B
0440 0444 2627 2627 /C/D
0441 0445 3031 3031 /E/F
0442 0446 3233 3233 /G/H
0443 0447 3435 3435 /I/J
0444 0450 3637 3637 /K/L
0445 0451 4041 4041 /M/N
0446 0452 4243 4243 /O/P
0447 0453 4445 4445 /Q/R
0450 0454 4647 4647 /S/T
0451 0455 5051 5051 /U/V
0452 0456 5253 5253 /W/X
0453 0457 5455 5455 /Y/Z
0454 0460 6621 6621 /LEFT,BRACK/BACK SLASH
0455 0461 7676 7676 /RT,BRACK/UP ARROW
0456 0462 7676 TABE, 7676 /BACK ARROW/N,U,
0457 /
0460 /INITIAL START 20
0461 0463 6474 ST20, JMP SETUP
0462 0464 6356 JMP INIT
0463 0465 6472 JMP GOMAN
0464 /
0465 /INITIAL START 400
0466 0466 6474 ST400, JMP SETUP
0467 0467 6356 JMP INIT /INITIALIZE TRAP PROCESSOR
0470 0470 0643 LDF 3
0471 0471 0000 HLT

```

0472	0472	0600	GOMAN, LIF	0	
0473	0473	6033	JMP	GUIDE	/THIS LANDS IN FIELD 0
0474			/TRAP PROCESSOR RELOCATOR		
0475	0474	0057	SETUP, SET	17	
0476	0475	0000		0	
0477	0476	1020	LDA I		/SET UP RESTARTS
0500	0477	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	0	/SET DATA FIELD TO 0
0505	0504	0061	SET I	1	/SET UP MOVE COUNT
0506	0505	7334	-TABE+17		
0507	0506	0062	SET I	2	/SET TO-ADDRESS
0510	0507	2017	2017		/BIT 1 (2-010) SAYS USE OF
0511	0510	0063	SET I	3	/SET FROM-ADDRESS
0512	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
0520	0517	2677	2677		/SWITCHES RESTART
0521	0520	0061	SET I	1	/20 JMP GUIDE-S
0522	0521	7757	-20		
0523	0522	1020	LDA I		
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	0527	6017	JMP	17	

0000 ERRORS

AC 4242
COFX 4171
DOKBD 4250
ENDX 4365
ERR 4130
FETCH 4160
GOMAN 4472
GRC 4262
GUIDE 4033
INIT 4356
KBD 0515
KCCA 6032
KRBA 6036
KRSA 6034
LC 4225
LOO 4323
LOO1 4327
LOW 4317
LP 4255
M70 4135
ONE 4136
OPR 0500
OP14 4212
OTHERS 4211
OVL 4222
OVN 4247
PC 4134
PRINT 4350
Q 4232
-

RET	4221
RETX	4337
RMFA	6244
RST400	4025
RTJ	4246
SETUP	4474
ST20	4463
ST400	4466
TAB	4401
TABE	4462
TLSA	6046
TSFA	6041
UPC	4252