SYMBOLIC TAPE EDITOR PROGRAMMING MANUAL

PDP-7

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

SYMBOLIC TAPE EDITOR

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

PREFACE

The program discussed in this manual, though written for use on the Programmed Data Processor-7 computer, can also be used without change on Digital's Programmed Data Processor-4. This compatibility between the libraries of the two computers results in three major advantages:

1. The PDP-7 comes to the user complete with an extensive selection of system programs and routines making the full data processing capability of the new computer immediately available to each user, eliminating many of the common initial programming delays.

2. The PDP-7 programming system takes advantage of the many man-years of field testing by PDP-4 users.

3. Each computer can take immediate advantage of the continuing program developments for the other.

CONTENTS

INTRODUCTION

Of all the necessary jobs confronting the programmer when debugging, perhaps the most tedious is making a corrected symbolic program. It involves seemingly endless duplication, listing, splicing, and close attention. The Symbolic Tape Editor for Programmed Data Processor-7 helps to lighten the task and speed up the process of correcting programs by using the computer to perform the drudgery.

In brief, the Editor reads a section of symbolic tape into core memory, where it is available for examination and correction. The corrected text can then be punched out on a new tape. Text may also be entered directly from the keyboard for original tape preparation. The Editor accepts tape input and provides output in either ASCII* or FIODEC codes. Tapes can be converted from one to the other using the Symbolic Tape Editor. In this description, it is assumed that ASCII tapes are used; where the treatment differs for FIODEC it will be noted. Keyboard communication with the Editor may be accomplished on either Teletype Model 33 KSR or Model 28 KSR. This manual assumes use of Model 33 KSR; Appendix 2 contains instructions for using Model 28 KSR where this differs from Model 33.

The information to be corrected is stored in a text buffer, which occupies all of memory not taken up by the Editor itself and has a capacity of approximately 4000 characters in a PDP-7 with 4096 words of memory, or approximately 16,000 characters in one with 8192 words.

HOW THE EDITOR WORKS

By convention, paper tape information is organized into various-sized blocks. The larger blocks are called pages and are separated by form feed codes on paper tape (FIODEC: stop codes). Each page is divided into smaller blocks called lines, which are separated by carriage return - line feed pairs (FIODEC: CR). A terminating carriage return, line feed pair is part of the line that precedes it.

---

*The code herein referred to as ASCII is actually #33 code or ASCII with the eighth bit punched on tape.

In the text buffer, lines are implicitly numbered decimally in sequence beginning with 1. Form feeds (FIODEC:: stop codes) are not stored in memory; hence there are no page divisions in the text buffer, and the entire contents of the Editor's buffer are treated as a single page. The user may organize his output into pages if he wishes.

Operating Modes

In order to distinguish between commands to itself and text to be entered into the buffer, the Editor operates in one of two modes. In Command mode, typed input is interpreted as directions to the Editor to perform some operation. In Text mode, all typed input is taken as text to be inserted in or appended to the contents of the text buffer. To help the user keep track of the mode, a visual indication is provided by the LINK light on the PDP-7 console. In Command mode, this light is off; in Text mode, it is on.

Commands

A command directs the Editor to perform some operation. A command consists of a single letter preceded by zero, one, or two arguments. If we let E represent any command letter, the three ways of constructing a command are:

No arguments	E)*
One argument	nE)
Two arguments	m,nE)

Note that two arguments must be separated by a comma, but that no comma separation is allowed between the argument(s) and the command.

To be executed, a command must be followed by a carriage return. This is the signal to the Editor to process the information just typed. The Editor responds with a line feed as soon as it has processed the command and begun the operation. If the Editor is doing something at the time a carriage return is typed (such as punching or reading tape), delay may occur before the line feed is returned. If a mistake is made while typing a command, the entire line may be ignored by typing Line Feed.

---

*The nonprinting characters carriage return and tab will be represented hereafter by) and →| respectively.

Arguments

An argument may be any decimal integer, special symbol, or arithmetic expression, consisting of decimal integers or special symbols separated by the addition (space) or subtraction (minus sign) operators.

Examples:

```
256
/-39
25-3
12 48-7
. 4
```

The following two characters are special symbols in the Editor.

/ This symbol represents the total number of lines in the text buffer. If there are 100 lines in the text buffer, the symbol / has a value of 100. It is used primarily for referencing the last line in the buffer.

Example:

/L) This will cause the last line in the text buffer to be listed.

The symbol period (.) has a value equal to the number of the last line involved in an Editor operation. The value of period will be equal to the number of the last line referenced or typed in except in the case of the delete command (D), where it will be equal to the number of the line preceding the first deleted line. In each command description that follows, the value of . after the completion of the operation is stated.

Example:

23,.C) Lines 23 to the current line are changed.

Special Functions

Certain keys have special operating functions. The first three below are nonprinting, so the symbol in parentheses is used to indicate the operation of the associated key.

Carriage Return (↵) — In both Command and Text modes, this is the signal for the Editor to process the information just typed. In Command mode, the operation specified is to be performed. In Text mode, it means that the preceding line of text is to be placed in the buffer.

Continuation (?↵) — In Text mode, this facilitates adding comments to successive lines or for end-of-line corrections. If a line of text is terminated by this pair instead of a single carriage return, the line will be entered as usual; then the line immediately following it will be printed up to but not including its carriage return. Thus, the new line is left open for additions or corrections. The editing example in the next section illustrates the use of ?↵. NOTE: If in text mode as a result of the Y command, continuation may not be used.

Line Feed (↓) — This character has two meanings depending on when it is used. If it is struck after some information has been typed, it causes that information to be deleted. Used thus in either mode, it has the effect of erasing mistakes. When it has processed the line feed, the Editor responds with a carriage return. If, in Command mode, line feed is the first character typed on a line, the next line of text (line .+1) will be printed.

Rub Out (RO) — This key has three distinct functions. Typing RO in Command mode will cause the next line of text to be printed. The use of RO for this purpose is preferred to that of line feed since it provides a neater printout.

Pressing RO in Text mode will cause the last character of an incomplete line of text to be deleted from the input buffer. Continued striking of this key will cause successive characters to be deleted one by one, working from the end of the line back to the beginning. In this way, a mistake can be corrected without having to retype the whole line.

Example: Instead of DAC PTEM, the following line was typed:

DAC CTE

To correct the line, RO is struck three times, erasing the last three letters in succession, E, T, and C. The correct text is then typed, and the resulting line appears on the teleprinter as:

DAC CTEPTEM

It is stored in the text buffer, however, in correct form, as:

DAC PTEM

In Text mode, the rub out key has another function. Typed immediately after a carriage return, it signals the Editor to return to Command mode. If one deletes all the characters in an incomplete line and then strikes RO one more time, the Editor will also return to Command mode. No keyboard response is provided by the Editor; but when it enters Command mode, the LINK light, which has been on while in Text mode, goes out.

Colon (:)

When this symbol is typed in Command mode, the Editor will print the decimal value of the argument that precedes it followed by a carriage return. It is frequently used for determining the number of lines of text in the buffer.

Example:

/: 57

or in determining the number of the current line:

.: 32

Tabulation (→|)

The bell key produces a tabulation when struck. However PDP-7 keyboards vary; some have tab hardware, others do not. A tab is automatic on the former, but the Editor spaces out the tabulation on those keyboards without automatic tabs. To indicate to the Editor which type of keyboard is being used, AC switch 1 should be set as follows:

ACS1	Down	Keyboard equipped with automatic tabulation.
	Up	Keyboard without automatic tab.

The foregoing applies only to input; the Editor spaces out all tabs when printing.

Parity Checking

When symbolic input from tape is in FIO DEC code, the Editor will check parity, if desired, under control of AC switch 17. When a parity error is encountered, a diagnostic is printed which gives the line number and the character code in error. For example:

PARITY IN LINE 23 CHAR IS 252

The offending character is automatically replaced by code 76_8, which is unused in the FIODEC set, and the Editor continues reading. During the editing process when a line containing a parity error is printed, the error code is printed as ?.

ASCII Input and Output

Tape format for input and output must be indicated by setting AC switches 15 and 16 as follows:

ACS15	Down	FIODEC output
	Up	ASCII output
ACS16	Down	FIODEC input
	Up	ASCII input

In addition when ACS15 is up, output control for ASCII is provided by AC switches 13 and 14.

ACS13	Down	Tape feeds are punched as 000
	Up	Tape feeds are punched as 200
ACS14	Down	ASCII tab code is punched
	Up	Tabs are translated into the number of spaces to effect proper formatting

USING THE EDITOR

Example 1

The following detailed example of the editing of a page of text will familiarize the reader with the basic operations of the Editor. All of the commands are described in full after the example, and the sample page of text is reproduced in Figure 1.

To begin, place the Editor program tape in the reader and load the program into memory.* The Editor starts automatically with cleared buffers. All subsequent operations, including loading the symbolic tape to be edited, are performed through the Editor from the teleprinter keyboard. Now set the AC switches as desired. Our sample text is a page from the symbolic tape of the Editor itself; it is punched in ASCII format. We wish to have ASCII output, and we assume that the keyboard has automatic tabulation. Our AC switch control settings are as follows:

ACS	1	13	14	15	16
	down	down	down	up	up

The Editor is now ready to read a symbolic tape or accept text from the teleprinter.

Whenever the Editor encounters an interrupt from a device other than the teleprinter, tape reader, or punch, it will halt with the I/O status word displayed in the AC lights. The user should examine the status word to determine the source of the interrupt, set the proper I/O command to clear the interrupt flag into the AC switches and depress CONTINUE to load it. The

*For a detailed description of how to load a binary object program tape, see the PDP-7 Assembler (Digital 7-3-S).

```
SYMBOLIC TAPE EDITOR PART 2 MAIN SUBROUTINES                                  1
                                                                              2
X1 = 10                                                                       3
X2 = 11                                                                       4
                                                                              5
APPEND,     0                                                                 6
            CHKONE                                                            7
                                                                              8
            SKP                                                               9
            JMP CONERR                                                       10
            LAC LASIN                                                        11
            DAC THISN                                                        12
            LAC LAST                                                         13
                                                                             14
            DAC THIS                                                         15
            LAC (NOP                                                         16
                                                                             17
            DAC TISW        /TELETYPE INPUT SWITCH                           18
            DZM LSTCHR      /STORAGE FOR LAST CHARACTER SEEN                 19
            LAC (JMS TONE   /SET SWITCH IN PACK ROUTINE                      20
                                                                             21
            DAC ON          /FROM TELETYPE                                   22
            JMS A           /APPEND ALL LINES AND RESET POINTERS             23
                                                                             24
            JMP I APPEND                                                     25
                                                                             26
A,          0                                                                27
            JMS PACK        /ADD ONE LINE                                    28
            JMP .-1                                                          29
            JMP I A                                                          30
                                                                             31
TONE,       0               /TYPE IN                                         32
            TTI                                                              33
            ISZ TONE        /INDEX RETURN IF NOT DONE                        34
            DAC LSTCRR      /SAVE CHARACTER FOR CHECK IF BLANK KEY TYPED      35
            JMP I TONE                                                       36
                                                                             37
                                                                             38
                                                                             39
BLANK,      LAC LSTCHR                                                       40
            SAD (77                                                          41
            JMP I TONE      /NO DELETE IF LAST CHARACTER WAS CR               42
            JMS DECR        /OTHERWISE ERASE ONE CHARACTER                   43
            JMP TONE + 1                                                     44
                                                                             45
DECR,       0                                                                46
            LAC PCNI        /IF CHARACTER CNT IS -2 MUST BACK UP 1 WORD       47
            SAD (-2                                                          48
            JMP FUR                                                          49
            ADD (-1         /OTHERWISE DECREMENT CHARACTER COUNT BY 1         50
            DAC PCNI                                                         51
            LAC PIEM        /GET PARTIAL ACCUMULATION                        52
DE1,        AND (777700     /AND OUT LAST CHARACTER                          53
            CLL                                                              54
            RTR     RTR     RTR                                              55
            DAC PIEM        /ROTATE WORD BACK ONE CHARACTER                  56
            JMP I DECR                                                       57
```

Figure 1 Editor Part 2 Main Subroutines

computer will halt immediately. The user should then reset the AC switches to their original control settings and depress CONTINUE again. The manually inserted I/O command will be executed and processing resumed. The Editor will clear the flags of the following devices upon loading:

Paper Tape Reader
Paper Tape Punch
Keyboard-printer
Clock
Mag Tape Type 57A
Line Printer Type 62
Card Reader
Card Punch
Display Light Pen
Character Generator

However, if the installation has a number of special I/O devices, the above operation may have to be performed a number of times after loading the Editor.

The Editor may be restarted at address 22 without affecting the contents of the buffer.

We are now ready to read in our symbolic text. The command for this operation is:

R)

This causes the Editor to read the input tape until a form feed (FIODEC: stop code) or the physical end of tape is encountered.

Once the text is in the buffer, we might wish to find out how many lines there are in the page, as follows:

/: 57

With this information, we can find our way about the page. Note that blank lines are included in the count.

The first task is to insert a comment before the line that begins with the symbol A. In order to do this, we must find the number of that line. This can be done by asking the Editor to print

out a line with the number we estimate is the correct one. We can see that the line is a little less than halfway down the page, and we know that there are 57 lines on the page, so we ask for line 25 using the command L, as follows:

```
25L)
        JMP I APPEND
```

Now we see from the program copy that the line we want is really two lines further on.

To insert text before the desired line, we will use the command 27I. This causes the Editor to enter Text mode. As many lines of text as we wish are then entered into the text before line 27. Having done so, we strike the RO key to return to Command mode. The result of our work:

```
27I)
→|     /THE FOLLOWING SUBROUTINES HANDLE)
→|     /INCOMING DATA.)
(RO)
```

If we wish, we can now verify our work by asking for a printout of the lines including the new text. Bear in mind that the inserted lines have been added; the buffer now contains 59 lines; and all lines after the insertion have had their numbers increased by two. Our verification would look like this:

```
25,29L)
        JMP I APPEND
        /THE FOLLOWING SUBROUTINES HANDLE
        /INCOMING DATA.
A,              0
```

Our next correction is to change the name of the subroutine PACK to LPAC. To do this, we use the Change command, C. From the previous printout, we know that the line beginning with the symbol A is now line 29, so the one we wish to change is now line 30. The operation is completed as follows:

```
30C)
    →| JMS LPAC          /ADD ONE LINE)
```

Suppose we wish to insert a comment following the next line in the buffer. Using the Continuation operator, we can do so easily, as shown:

```
30C)
        →| JMS LPAC        →|/ADD ONE LINE?)
        →| JMP .-1
```

Now we add the comment and return to Command mode.

```
30C)
        →| JMS LPAC        →|/ADD ONE LINE?)
        →| JMP .-1         →|/GET NEXT LINE)
(RO)
```

If we have no further corrections to make, we are ready to punch out the corrected text. The following sequence of commands causes the contents of the buffer to be punched, followed by some tape feed and a stop code. Finally, the contents of the buffer are totally erased.

```
P)
S)
K)
```

We are now ready to read the next page of the input tape. Figure 2 shows the whole of our operations for this example; the comments in parentheses summarize what happens.

Rearranging Text on Output

Text may be rearranged on output by using the nP and n,mP commands to punch lines or groups of lines from the Editor's buffer in the order of their selection by the user rather than their order in the buffer. The buffer contents are unaffected by this process.

For example, if we wished to relocate the six lines beginning at line 40 in the sample text at the end of the page, we would type:

1,39P)	to punch the first section of text
46,/P)	to punch the final section of text
40,45P)	to relocate the desired lines to the end of the tape being punched
S)	to complete the page with a stop code and tape feed

```
R)                                          (read one page of input tape)
/:      57                                  (find number of lines)
25L)                                        (print line 25)
        JMP I APPEND
27I)                                        (insert text before line 27)
    /THE FOLLOWING SUBROUTINES HANDLE)
    /INCOMING DATA.)
(RO)25,29L)                                 (return to Command mode and
        JMP I APPEND                        verify previous insertion)
    /THE FOLLOWING SUBROUTINES HANDLE
    /INCOMING DATA
A,      0
30C)                                        (change line 30)
        JMS LPAC    /ADD ONE LINE?)         (continuation; remain in
        JMP .-1     /GET NEXT LINE)         text mode)
(RO)P)                                      (return to Command mode and
S)                                          punch out corrected text.)
K)                                          (erase the text)
```

Figure 2 Editing Example 1

Example 2

This example illustrates some of the more subtle commands available to the Editor user. It is suggested that this section not be studied until the user has become acquainted with the fundamental aspects of the Editor illustrated in Example 1.

To begin, we return to the sample text as given in Figure 1 and read the page into the buffer.

```
R)
```

Suppose now we wish to make an addition to a line following symbolic location BLANK. To find its line number in the buffer, we will use the nG command. This command searches from line n until a line is found which does not begin with a tab, carriage return, or slash and prints the line. To avoid stopping at earlier symbolic addresses, we choose n as 35.

```
35G)
BLANK,      LAC LSTCHR
```

To find the line number, we use the character period.

```
.:40
```

We know the line we are concerned with is two numbers further, that is line 42. In line 42, we wish to change the instruction JMP I TONE to JMS TONE deleting the I and changing the P to S. To accomplish this, we use the command nY. This causes line n to be typed slowly and allows interruption with the rub out (RO) and alteration of the printed text. A carriage return will append the unprinted portion of the line of text to the buffer.

```
42Y)

        JMP I(RO)
```

This rub out (RO) puts the Editor into text mode. Now type the insertion or delete using the rub out key:

```
JMP I(RO) (RO) (RO) (RO) S
```

The first rub out put the Editor into Text mode; the second rub out deletes the I, the third rub out deletes the space, the fourth rub out deletes the P. Now the S is added, and a carriage return appends the unprinted portion of the line to the buffer. We list the line to check our corrections.

```
42L)

        JMS TONE     /NO DELETE IF LAST CHARACTER WAS CR
```

To modify another portion of this line, it would be necessary to repeat the nY command. To inspect the altered line without bothering to print the comments, we could use the command nQ. This suppresses all text following the second tab on each line.

```
42Q)

        JMS TONE
```

Now suppose we want to punch the corrected text but move the instructions beyond the address DECR to the next page of our tape. We can punch the first portion of text down to line 46, then delete it moving DECR to line 1.

```
1,45P)
S)
1,45D)
1L)
DECR,)                          0
```

We may now read the next page of tape into the buffer following the instruction JMP I DECR. Figure 3 summarizes this example.

```
R)                                  (read 1 page of input tape)
35G)                                (print first symbolic addressed line after 35)
BLANK,      LAC LSTCHR
.:40                                (print the number of the current line)
42Y)                                (print line 42 slowly for corrections)
            JMP I(RO) (RO) (RO) (RO) S)
42L)                                (check correction made)
            JMP TONE                /NO DELETE IF LAST CHARACTER WAS CR
42Q)                                (check corrections, suppress comments)
            JMP TONE
1,45P)                              (punch lines 1 through 45)
S)                                  (punch a stop code)
1,45D)                              (delete lines 1 through 45)
1L)                                 (check the location of DECR)
DECR,       0
```

Figure 3 Editing Example 2

THE EDITOR'S COMMAND REPERTOIRE

The commands are grouped under four headings:

Input Tape Operations
Editing
Text Examination
Output (Punching) Operations

The following rules must be obeyed:

If a command requires one or two arguments, they must be provided. If an argument is missing, the Editor prints the error message,

ARG MISSING

and ignores the command as typed.

If an argument is greater than the final line in the buffer, the Editor prints

?

and ignores the command.

If a command does not take an argument and one is provided, the Editor simply types

?

and ignores the command. Note that some commands can legitimately take all three forms: zero, one, or two arguments.

If a character not in the command repertoire is typed, the error message is again,

?

and the command is ignored.

If, while adding text to the buffer, either by reading or editing, the contents come within 200 characters of the storage capacity, the Editor interrupts at the end of the line being entered with the message,

BUFFER ALMOST FULL

If the programmer persists in adding text after this message, the Editor will interrupt a second time when there is no more room with the message,

BUFFER IS FULL

At this point, no more text may be inserted in the buffer. Some commands may still be used, however, such as examination or deletion, and the buffer may be punched.

Input-Tape Operations

In these commands, the form feed (FIODEC: stop code) and the physical end of the input tape are both end-of-page indicators. If an end of tape is encountered before the completion of a command, the operation is terminated.

Command	Action
R	Read a page of text. The Editor will read information from the input tape until a form feed (FIODEC: stop code) or the physical end of tape is encountered. The incoming text is appended to the contents of the buffer; no information in the buffer is lost. The form feed is not entered into the buffer.
nR	Read n lines (not pages) of text. The Editor will read n lines and append them to the contents of the buffer. Reading will cease if a form feed (FIODEC: stop code) or the end of tape is encountered before n lines have been read.
Z	Skip one page of text. The input tape will be moved forward until a form feed (FIODEC: stop code) is encountered. The contents of the buffer are unaffected.
nZ	Skip n pages of text. The contents of the text buffer are unaffected.

Editing

The following commands permit the alteration of text in the Editor's buffer.

Command	Action
nD	Delete line n. Line n is removed from the text buffer. The numbers of all lines following it are reduced by one, as is the line count. .=n-1 after execution.
n,mD	Delete lines n through m, inclusive. The line following line m becomes the new line n. .=n-1 after execution.
A	Append. The Editor enters Text mode upon processing this command, and the user may then type in any number of lines of text. These will be appended to the end of the text in the buffer or placed into a previously empty buffer. The line count is increased accordingly. This command may be given with an empty buffer to enter text. .=/ after execution.
nI	Insert text before line n. The Editor enters Text mode to accept input. No information is lost. The first line typed becomes the new line n. The numbers of all lines following the insertion, as well as the line count, are increased by the number of lines inserted. If k lines are inserted, then .=n+k-1 after execution.
nC	Change line n. Line n is deleted, and the Editor enters Text mode to accept input. The user may now insert as many lines of text as he wishes in place of the deleted line. If more than one line is inserted, subsequent lines will be renumbered. If k lines are inserted, .=n+k-1 after execution.

Command	Action
n,mC	Change lines n through m. These lines (inclusive) are deleted. The user may insert any number of lines (or none at all) in their place. Period will be equal to the number of the last line typed in.
n,mX	External insertion. The Editor will read the next n lines of text from the symbolic tape in the reader and insert them after line m in the text buffer. If a form feed (FIODEC: stop code) is encountered before n lines have been read, reading will stop. Period will be equal to the number of the last line inserted.
nY	Type line n at reduced speed (1 character/second) to allow for correction of individual characters within a line. Striking the rub out key after any character will suspend output and put the Editor in Text mode. The user may then delete text using the rub out key or add text. Typing a carriage return causes the unprinted text remaining in the line to be appended to the corrected buffer and the Editor to be returned to the Command mode. .=n after execution.
K	The contents of the buffer are completely erased. The values of / and . are set to zero.

Text Examination

The following commands will cause the printout of all or any part of the contents of the text buffer. Printing may be stopped at any time by the use of AC switch 0. Normally, this switch is down. If it is turned on then off at any time during a printout, the operation will stop immediately. The line being printed is unaffected in the buffer. If this occurs in the middle of a line, the user must type a carriage return to restore the Editor to normal Command mode

operation. Execution of any of the following commands may be halted in this manner. The contents of the text buffer are unaffected by the following operations unless specified.

Command	Action
nL	Print line n. This line will be typed out, followed by a carriage return and a line feed. .=n after execution.*
n,mL	Print lines n through m, inclusive. .=m after execution.*
W	Write. This causes the Editor to print the entire text contained in the buffer. The buffer remains intact. .=/ after execution.
nW	Write n pages. The text buffer is cleared, and the Editor reads n pages from tape, printing each one on a separate page, spacing across page perforations automatically. This command is equivalent to executing K, followed by the sequence, R, W, K, performed n times or until a physical end of tape is encountered. The buffer will be empty upon completion of this command.
Q	Uncommented print. The Editor will print the entire contents of the buffer, suppressing all text on each line after the second tabulation. Normally, this has the effect of suppressing comments in the program. .=/ after execution.
nQ	Print line n uncommented. Line n will be printed up to the second tabulation; all information following this is not printed. .=n after execution.
n,mQ	Print lines n through m uncommented. .=m after execution.

*Whether printing is stopped or not.

Command	Action
nG	Get and print next location symbol. The Editor will scan the text beginning with line n until it encounters a line beginning with a character other than), ↓, →|, or / . This line is printed. Period will be equal to the number of the line printed.
B	Back up and print. Line .-1 will be printed. The value of period is decreased by one after execution.

Punching Operations

The following commands provide for the output of corrected text or for the duplication of pages of the input tape. As with reading, punching can be halted by the use of AC switch 0. In this case, however, the switch must be kept up until processing of the punch command is finished. This is readily apparent by watching the AC lights. While processing, the AC is active; as soon as the operation is finished, the AC lights go out. ACS0 may then be turned off, and the Editor is ready to accept the next command.

Command	Action
P	Punch the entire contents of the text buffer. This is preceded by a short length of tape feed. The contents of the text buffer are unaffected by this command.
nP	Punch line n. This is preceded by a short length of tape feed. .=n after execution.
n,mP	Punch lines n through m , inclusive. .=m after execution.
S	Punch a form feed (FIODEC: stop code). This is preceded and followed by a short length of tape feed.

Command	Action
nF	Feed n lines of tape. The Editor will punch n lines of tape feed.
O	Punch one page. The Editor will punch the contents of the text buffer followed by a form feed (FIODEC: stop code) and then clear the buffer. This command is equivalent to the sequence: P, S, K.
N	Punch, then read the next page. The Editor will punch the contents of the buffer and a form feed (FIODEC: stop code) clear the buffer, and read the next page of tape into the buffer. This command is equivalent to the sequence: O, R.
nN	Punch, duplicate, and read. The Editor will punch the contents of the buffer, punch a form feed (FIODEC: stop code) clear the buffer, duplicate n-1 pages of tape, and then read the nth page into the text buffer. This command is equivalent to the sequence: P, S, (n-1) T, R
nT	Tape duplication. The Editor will clear the buffer, then read and punch n pages of tape. The buffer is empty upon completion of this command, which is equivalent to the sequence: K, n(R, P, S, K) If, within the range of this command, the Editor encounters two form feeds (FIODEC: stop codes) with nothing more than tape feed between them, only one will be punched. However, the Editor will count the space between them as a separate page in reading the input tape.

APPENDIX 1

SUMMARY OF EDITOR OPERATIONS

AC Switch Settings

Switch		Function
0	Down	Normal operation.
	Up	Stop printing or punching
1	Down	Teletype with automatic tabulation.
	Up	Teletype without automatic tab.
13	Down	ASCII tape feeds punched as 000.
	Up	ASCII tape feeds punched as 200.
14	Down	ASCII output: transmit tab characters.
	Up	Convert tabs into proper number of spaces.
15	Down	Punch output in FIODEC code.
	Up	Punch output in ASCII code.
16	Down	Input tape is in FIODEC code.
	Up	Input tape is in ASCII code.
17	Down	Check parity on FIODEC input tape when reading.
	Up	Ignore parity errors.

Special Key Functions

rub out key (text mode)	Leave text mode (if first character typed); otherwise, erase last character.
rub out key (y command)	Stop output.
rub out key (command mode)	Print next line.
line feed	Print next line (if first character typed); otherwise delete all typed input.
carriage return	Complete specified action.

Editor Command Summary

Command	Arguments	Function
A	0	Append.
B	0	Back up and print.
nC	1	Change line n.
n,mC	2	Change lines n through m.
nD	1	Delete line n.
n,mD	2	Delete lines n through m.
nF	1	Feed n lines of tape.
nG	1	Get next location tag after line n.
nI	1	Insert text before line n.
K	0	Kill the text buffer.
nL	1	Print line n.
n,mL	2	Print lines n through m.
N	0	Punch and Next page. Equivalent to P, S, K, R.
mN	1	Punch, duplicate, and read. Equivalent to P, S, m-1 (T), R.
O	0	Punch and kill. Equivalent to P, S, K.
P	0	Punch the contents of the buffer.
nP	1	Punch line n.
n,mP	2	Punch lines n through m.
Q	0	Print uncommented -- entire buffer.
nQ	1	Print line n uncommented.
n,mQ	2	Print lines n through m uncommented.
R	0	Read one page of text.
nR	1	Read n lines of tape.
S	0	Punch form feed (FIODEC: stop code).
nT	1	Duplicate n pages of tape. Equivalent to K, n(R, P, S, K).
W	0	Write the entire buffer.
nW	1	Write n pages. Equivalent to K, n(R, W, K).
n,mX	2	External insert. Insert n lines of text from tape after line m.
nY	1	Individual character correction on line n.
Z	0	Skip one page of text.
nZ	1	Skip n pages of text.

APPENDIX 2

USING THE EDITOR WITH TELETYPE MODEL 28 KSR

Special functions (see page 4).

1. Continuation - use $) in place of ?).

2. Use BLANK key in place of RUB OUT key.

(On Model 28 keyboards, it is necessary to hold down the UNLK key if BLANK is to be repeatedly struck, otherwise the keyboard will lock.)

Parity checking (see page 6).

The parity error code is $.

NON-EQUIVALENT CHARACTERS

Regardless of the format of input or output tapes, the Editor stores all text internally in FIODEC Concise Code (6-bit characters). Any FIODEC code which does not have a standard equivalence on the Type 28 Teleprinter Keyboard, as specified in Appendix 1, may be placed into text by using the format

$abc

where the characters bc represent the 2-digit octal code. The first character a must be 0 if the code is to be lower case; 1 if in upper case. Such a code will appear in the same format when printed by the teleprinter. However, the code will appear on FIODEC output tapes as the character whose code is bc in the proper case.

Example: There is no equivalent character combination on the Type 28 Teleprinter Keyboard for the FIODEC backspace. Consequently, in order to overprint an O with a slash, the user would have to type O$075/ on the teleprinter. This character would appear in the buffer as O (letter), backspace, / (slash). If a tape were punched containing this sequence and read by a Flexowriter, the result would be Ø.

CHARACTER SET EQUIVALENCES

The Teletype Model 28 KSR character set, represented by a 5-bit code, is considerably smaller than either the ASCII (7-bit) or FIODEC (6-bit) sets. As a result, it is necessary to use an escape character in order to represent all the characters of the latter two sets when using the Teletype 28 keyboard and printer. This escape character is $. It means that the character immediately following the $ is to be interpreted differently than it would if it appeared alone. In effect, the $c pair (where c represents any Teletype 28 character) is interpreted as representing a single ASCII or FIODEC character. Appendix 3 gives the equivalences among the three sets.

APPENDIX 3

CHARACTER SET EQUIVALENCES

ASCII, Teletype 33 KSR	FIODEC		Teletype 28 KSR
0-9	0-9		0-9
A-Z	a-z		A-Z
A-Z	A-Z		$A-$Z
.	.	(period)	.
,	,		,
-	-	(minus sign)	-
/	/	/	/
:	:	(center dot, period)	:
;	;	(center dot, comma)	;
(	(		(
)	)		)
+	+		&
↑	↑		!
*	x	(multiply)	#
"	"		$"
'	'		$'
=	=		$:
[	[		$(
]	]		$)
<	<		$-
>	>		$&
←	⊂		$?
%	⊃		$,
!	∨		$/
&	∧		$#
RO	→	(blank)	$;
\	|	(vertical stroke)	$!
$	_	(underbar)	"
no equivalent	·	(center dot)	$.
#	¯	(overbar)	'
Form Feed	Stop Code		n.e.
↓	↓		↓
Tab	Tab		bell
?	n.e.		$,?

digital
EQUIPMENT
CORPORATION
MAYNARD, MASSACHUSETTS

PRINTED IN U.S.A.

10-7/66