

NUCLEAR DATA, INC.
Post Office Box 451
Palatine, Illinois 60067

November, 1972

IM41-0002-04
SOFTWARE INSTRUCTION MANUAL
ND812 SYMBOLIC TEXT EDITOR

Copyright 1972, by Nuclear Data, Inc.
Printed in U.S.A.

"THIS DOCUMENT IS THE EXCLUSIVE PROPERTY OF NUCLEAR DATA, INC. AND MAY NOT BE REPRODUCED, NOR MAY THE INFORMATION CONTAINED THEREIN OR DERIVABLE THEREFROM BE USED IN ANY MANNER, EXCEPT BY WRITTEN PERMISSION OF NUCLEAR DATA, INC. THE PROPRIETARY RIGHTS TO THE AFORESAID INFORMATION, BOTH OF A PATENTABLE AND UNPATENTABLE NATURE, ARE EXPRESSLY RESERVED TO NUCLEAR DATA, INC."

TABLE OF CONTENTS

| <u>SECTION</u> | <u>TITLE</u> | <u>PAGE</u> |
|----------------|--|-------------|
| I | INTRODUCTION | 1-1 |
| | 1.1 General | 1-1 |
| | 1.2 Program Area | 1-1 |
| | 1.2.1 Program Execution Area | 1-1 |
| | 1.2.2 Program Buffer Area | 1-1 |
| | 1.3 Starting Address | 1-1 |
| | 1.4 Equipment Configuration | 1-1 |
| | 1.4.1 Minimum Equipment | 1-1 |
| | 1.4.2 Optional Equipment | 1-2 |
| | 1.5 Definitions | 1-2 |
| | 1.5.1 Source Program/Tape | 1-2 |
| | 1.5.2 Symbols | 1-2 |
| | 1.6 Related Publications | 1-2 |
| II | PROGRAM DESCRIPTION | 2-1 |
| | 2.1 Introduction | 2-1 |
| | 2.2 Command Mode and Text Mode Description | 2-1 |
| | 2.3 Editor Functional Description | 2-1 |
| | 2.3.1 Lines and Blocks | 2-2 |
| | 2.3.2 Page Divisions | 2-2 |
| III | OPERATIONAL PROCEDURE | 3-1 |
| | 3.1 Introduction | 3-1 |
| | 3.2 Loading a Symbolic Tape Using the Editor | 3-1 |
| | 3.3 Editing a Symbolic Tape | 3-2 |
| | 3.4 Writing New or Corrected Symbolic Tape | 3-3 |
| | 3.5 New Source Programs | 3-3 |
| | 3.5.1 Generating a New Source Program | 3-4 |

| <u>SECTION</u> | <u>TITLE</u> | <u>PAGE</u> |
|----------------|---|-------------|
| IV | OPERATOR OR USER CONTROL | 4-1 |
| 4.1 | Introduction | 4-1 |
| 4.2 | Special Characters and Functions | 4-2 |
| 4.2.1 | Carriage Return | 4-2 |
| 4.2.2 | Left Arrow. | 4-2 |
| 4.2.3 | Rub-Out. | 4-2 |
| 4.2.4 | Form Feed | 4-3 |
| 4.2.5 | Period | 4-3 |
| 4.2.6 | Slash | 4-3 |
| 4.2.7 | Line Feed | 4-3 |
| 4.2.8 | Left Angle Bracket | 4-4 |
| 4.2.9 | Colon | 4-4 |
| 4.2.10 | Blank Tape and Leader/Trailer | 4-4 |
| 4.2.11 | Tabulation. | 4-4 |
| 4.3 | ND812 Switch Register Input/Output Device Select Options | 4-5 |
| 4.3.1 | Input Device Select Bits 8 and 9 | 4-5 |
| 4.3.2 | Output Device Select Bits 10 and 11 | 4-6 |
| 4.3.3 | Keyboard/Printer Interrupt. | 4-6 |
| 4.4 | Command Description | 4-6 |
| 4.4.1 | Input Commands | 4-7 |
| 4.4.2 | Output Commands | 4-8 |
| 4.4.3 | Editing Commands | 4-10 |
| V | COMMAND SUMMARY. | 5-1 |
| 5.1 | Introduction | 5-1 |
| 5.2 | Special Characters | 5-1 |
| 5.3 | Edit Commands. | 5-2 |
| APPENDIX | | |
| | Appendix A - Loading and Initialization of the ND812 Text Editor. | A-1 |

SECTION I INTRODUCTION

1.1 GENERAL

This manual provides instructions for using the ND812 Symbolic Text Editor ND41-0002-04 (hereinafter referred to as the Editor).

The Editor allows the user to prepare and edit source tapes on line in ASCII code with the Teletype and/or High Speed Reader/Punch or Tape Cassette. The tedious task of correcting symbolic program tapes using the teletype off-line is thereby avoided. Proper use of the ND812 Symbolic Editor can substantially ease the labor and reduce the number of passes necessary to produce correct source program tapes.

1.2 PROGRAM AREA

1.2.1 PROGRAM EXECUTION AREA

ND812 Memory locations 0050_g through 3507_g are used for program execution.

1.2.2 PROGRAM BUFFER AREA

ND812 memory locations 3510_g through 7464_g are used as a text buffer.

1.3 STARTING ADDRESS

The starting address of the Text Editor is 0200_g.

1.4 EQUIPMENT CONFIGURATION

1.4.1 MINIMUM EQUIPMENT

The minimum equipment required for proper operation of this program is:

- a. 4K ND812 Central Processor.
- b. TC33ASR Teletype (low speed punch/reader).

1.4.2 OPTIONAL EQUIPMENT

Optional equipment supported by this program includes:

- a. High speed punch/reader.
- b. Magnetic tape cassette unit.
- c. Line printer.

1.5 DEFINITIONS

1.5.1 SOURCE PROGRAM/TAPE

The source program is written by the programmer in the symbols allowed in the BASC-12 symbolic language, and entered into the Editor buffer area via keyboard or magnetic tape cassette. The program is edited on line by the programmer via keyboard commands, and the output is either a first generation or corrected source program tape (paper tape or cassette).

1.5.2 SYMBOLS

A symbol consists of any collection of letters, numbers, or other characters appearing on a line, not in a comment, and followed by a terminator (space).

1.6 RELATED PUBLICATIONS

Related publications include the following.

1. IM41-0000-00, Principles of Programming the ND812 Computer in Assembly Language.
2. IM41-0001, Software Instruction Manual, BASC-12 General Assembler.
3. IM41-0028, Software Instruction Manual, BASC-12 General Assembler (8K) For Line Printer Printout.
4. IM41-0084, Software Instruction Manual, BASC-12 General Assembler (8K) For Teletype Printout.

SECTION II PROGRAM DESCRIPTION

2.1 INTRODUCTION

This section provides a brief description of the Editor. The Editor provides two modes of operation which allows a programmer to prepare and edit source programs/tapes on line in ASCII code via Teletype, and/or High Speed Reader/Punch, or Cassette Transport.

The two modes of Editor operation are Command Mode, and Text Mode. The Command Mode allows manipulation of source program or text stored in the text buffer. The Text Mode allows text to be replaced, inserted into, or appended to the contents of the text buffer via teletype keyboard. New source programs may also be generated in Text Mode via Teletype keyboard.

2.2 COMMAND MODE AND TEXT MODE DESCRIPTION

The Editor provides three types of commands. These are Input Commands, Output Commands, and Editing Commands (refer to Section IV for command descriptions). The Editor operates in either Command Mode or Text Mode to distinguish between editing commands, and actual text which is entered into the text buffer. In Command Mode, all input typed on the teletype will be interpreted as commands to the Editor to perform some operations, or to allow the operator to perform some operation on the text stored in the buffer. In Text Mode, all typed input is interpreted as text to replace, be inserted into, or to be appended to the contents of the text buffer (the Editor is in Command Mode after it is initially loaded into core memory). The user types the desired command code and terminates it by striking the carriage-return key. This non-printing character for carriage-return is hereinafter represented by (↵) and tells the Editor to carry out the command. The Editor then enters Text Mode and responds with a line-feed character represented by the up arrow (↑) as soon as it has processed the command and has begun the operation. With the Editor in Text Mode, he enters a form-feed (CTRL/FORM combination) to tell the Editor to return to Command Mode. The Editor answers with a carriage return/line feed to indicate the transition back to Command Mode.

2.3 EDITOR FUNCTIONAL DESCRIPTION

The read Input Command directs the Editor to read a block of source tape (via Teletype,

High Speed Reader, or Tape Cassette) into a buffer in memory where it is available for examination and corrections. The block buffer occupies 2048 memory locations and has a capacity of exactly 4095 characters. When the Editor is finished reading a block into the buffer, a carriage return/line feed signals the user that he may begin editing. The user may then call for a listing of individual numbered lines in any order and insert desired changes and corrections. In addition, text may be added to the buffer or inserted between specified lines. Groups of lines or individual lines may be moved or deleted by a single command or the entire buffer may be erased if desired. Upon command, the Editor will either list the corrected lines of the block on Teletype, or write them on paper tape or magnetic tape. The Editor can also be used to generate a new source program/tape in Text Mode by typing new text directly on the Teletype keyboard.

The Editor is an independent program and has a starting address of 0200₈.

2.3.1 LINES AND BLOCKS

By convention, tape information is organized into lines in variable size lines and blocks. Blocks are separated on tape by an ASCII form-feed code and lines are separated by carriage returns. A block of text residing in the buffer at one time may contain about 60 lines of heavily commented text (4095 characters) or 340 lines without comments or formatting. Each line of text includes the terminating carriage return. All lines in the text buffer are implicitly numbered in decimal notation starting with 1. This implicit enumeration is continually updated by the Editor to take account of the line insertions, moves and deletions. For editing and listing purposes, each line is referred to by its current implicit decimal number on the page. The form-feed character is not stored, but is generated on output at the user's discretion.

2.3.2 PAGE DIVISIONS

Page divisions may be entered by typing a CTRL/K combination. Although the entire contents of the buffer are treated as a single block during editing, the user may output the contents of the text buffer as several blocks, by writing the desired portions and using the "F" command between the new blocks.

SECTION III OPERATIONAL PROCEDURE

3.1 INTRODUCTION

This section describes the sequence of operations necessary to load, edit and write a corrected symbolic program tape, and provides brief examples of the use of the Editor. Loading and initialization of the Editor is described in Appendix A.

After the Editor has been loaded, it may be used to read a block of the symbolic program to be corrected into the text buffer. When the block has been read in, and a form-feed code encountered, the Editor generates a carriage return/line feed signaling the user that the Editor is in Command Mode. Corrections and additions may then be either typed on the Teletype keyboard or inserted from paper tape via the low speed reader. Individually numbered lines may be listed in any order permitting insertions, deletions or changes. Text may be inserted between specified lines, moved or appended to the end of a section. Individual lines, groups of lines or an entire page may be deleted on a single command. To insure that a tape is correct, desired portions or a desired block of the program may be listed before punching. Finally, the corrected lines, groups of lines, or the entire block may be punched and/or written. The original text remains available in the core buffer should further corrections be necessary. The following paragraphs provide detailed procedures for loading a symbolic source tape, making required corrections, and punching the corrected symbolic tape. Also, a brief procedure is provided for generating a new source program via Teletype keyboard.

3.2 LOADING A SYMBOLIC TAPE USING THE EDITOR

- a. Load the Text Editor as described in Appendix A if necessary.
- b. Depress ND812 STOP key. Set Switch Register to 0200_g.
- c. Depress LOAD AR and START keys.
- d. Clear buffer by typing K (carriage return/line feed).
- e. Set switches as indicated in Section IV, paragraph 4.3 - Switch register Options.

- f. Place the symbolic source tape of the program to be corrected in the appropriate input device.
- g. Enter the READ Command followed by a carriage return (↵) from the keyboard (i.e., type "R↵"). If the teletype reader is used, it should be turned on. The symbolic tape will then be read into the text buffer.
- h. The Editor will continue reading tape until the form-feed code is encountered at the end of the tape. If the tape contains no form-feed code, and the Teletype reader is being used for input, strike the CTRL/FORM key combination after the tape has been read in. Upon recognizing the form-feed character, the Editor enters the Command Mode and generates a carriage return/line feed to indicate that it is ready for the first command. If the low-speed paper tape reader is in use, it is preferable, though not necessary, that it be placed in the STOP position while editing. The first block of text is now ready for editing.

3.3 EDITING A SYMBOLIC TAPE

The actual editing procedure depends of course upon a particular user's requirements. The general procedure is illustrated in the example that follows.

- a. Load the Editor, and then the source tape as necessary per paragraph 3.2, steps a, b, and c. The Editor is now ready to receive editing commands. Terminate each command to the Editor by striking the RETURN (↵) key. This directs the Editor to execute the command.
- b. After a command to Insert, Change, or Append text to the text buffer has been executed, the Editor remains in the Text Mode until the operator types the CTRL/FORM key combination on the keyboard. This combination generates the form-feed character which tells the Editor to return to the Command Mode.
- c. If a great deal of text is added during the course of editing, the text buffer may overflow (full capacity is approximately 60 lines of heavily commented text or 340 lines of uncommented text). When the buffer storage limit has been exceeded, the program will print an exclamation point on the teleprinter and return to Command Mode. Editing may be continued by writing out and deleting a portion of the text which will free a corresponding number of locations for continued editing. A form-feed may also be generated to create a block division for future editing.
- d. The Editor may be stopped at any time by pressing the STOP key. To continue, depress the CONT key. If it is desired to restart the Editor without disturbing the buffer, place octal address 0200₈ in the Switch Register and depress the LOAD AR and CONT keys.

For input, editing, and output commands that may be given to the Editor, refer to the detailed explanation of the command structure, command repertoire, and special characters and functions under command description (Section IV); or refer to the corresponding summaries of commands and special characters in the Command Summary (Section V).

3.4 WRITING NEW OR CORRECTED SYMBOLIC TAPE

The procedure for punching or writing the new or corrected symbolic tape naturally depends on the user's requirements. The general sequence of steps is given below.

- a. Use the Write Command (refer to Section IV) to write one line of the text, lines m through n or the entire text.
- b. Before entering the final carriage return, be sure that the ND812 SWITCH REGISTER Bits 10 and 11 have been properly set up to specify the required output device (refer to paragraph 4.3), that the device has been write-enabled, and is ready to produce the output tape.
- c. Enter the final carriage return to start writing.
- d. If using the teletype punch, after a tape has been punched, turn the punch off before typing any further commands. If this is not done, the control codes typed will be punched on the symbolic tape.
- e. Punching or writing the symbolic program on tape does not delete it from memory; the block remains in the text buffer until the command is given to erase the first through last lines.

NOTE

The recommended block length as delineated by form-feeds is approximately 120 lines. However, the Editor will accept more text if necessary. It is most convenient, of course, to insert block divisions at major logical divisions of the program.

3.5 NEW SOURCE PROGRAMS

The Editor may also be used for generating a new source program on line by entering text directly on the Teletype keyboard. When writing a new source program in BASC-12 language, it is imperative that the user have a solid knowledge of the source language. A source program consists of mnemonic codes and characters arranged in a format acceptable by the BASC-12 Assembler. The following example illustrates the statement format which can be interpreted by the BASC-12 Assembler.

| <u>LABEL</u> | <u>INSTRUCTION</u> | <u>ADDRESS OR OPERAND</u> | <u>/COMMENT</u> |
|--------------|--------------------|-------------------------------|-------------------------|
| READ | JMP | END | /INPUT ONE CHARACTER |

Refer to Principles of Programming the ND812 Computer in Assembly Language, and the BASC-12 General Assembler Manuals (referenced in paragraph 1.6) for specific information on preparation of source programs.

3.5.1 GENERATING A NEW SOURCE PROGRAM

The procedure for generating a new source program at the Teletype keyboard is given below.

- a. Load the Text Editor as described in Appendix A.
- b. Depress ND812 STOP key. Set SWITCH REGISTER to 0200g.
- c. Depress LOAD AR and START keys.
- d. Set ND812 SWITCH REGISTER Bit "0" to "1" (refer to paragraph 4.3).
- e. Enter the APPEND Command followed by a carriage return (↵) on the Teletype keyboard (i.e., type "A↵"). The Editor is now in Text Mode, and the operator may enter any source program on the Teletype keyboard. The following is an example source program.

| <u>LABEL</u> | <u>INSTRUCTION</u> | <u>OPERAND</u> | <u>/COMMENT</u> |
|--------------|--------------------|----------------|-----------------------------|
| *200, | | | /ORIGIN OF PROGRAM |
| | CLB | JK | |
| | JPS | SUB | |
| | JMP | •-2 | |
| SUB, | 0 | | /SUBROUTINE ENTRY POINT AND |
| | ADJ | SUB1 | /RETURN ADDRESS RETAINER |
| | ADJ | P1 | |
| | STJ | SUB1 | |
| | JMP0 | SUB | /RETURN VIA SAVED ADDRESS |
| SUB1, | 0 | | /AT SUB ENTRY POINT |
| P1, | 1 | | /DATA STORAGE WORD |
| \$ | | | /CONSTANT STORAGE WORD |

NOTE

To obtain neat columns as above, simultaneously depress CTRL and I keys at Teletype. The Editor produces "tab stops" at 8 space intervals.

- e. After the source program has been entered into the text buffer, simultaneously depress Teletype CTRL and FORM key. The Editor is now in Command Mode. The operator may now edit the source program as necessary per paragraph 3.3 or write a symbolic tape per paragraph 3.4. Refer to Section IV for description of various Input, Output, and Editing commands available to the operator.

SECTION IV OPERATOR OR USER CONTROL

4.1 INTRODUCTION

This section describes special characters and functions for operation of the Editor, Input/Output device select Options, and Command Descriptions for the Editor.

A command directs the Editor to perform a desired operation. Each command consists of a single letter preceded by 0, 1, 2 or 3 arguments. The command letter tells the Editor what to do. The arguments usually specify which numbered line or lines of the text are affected (some arguments consists of special characters). Commands to the Editor must take one of the following forms, where "S" represents any command letter and the symbol (↵) represents a carriage-return.

| <u>Type of Command</u> | <u>Command Format</u> | <u>Meaning</u> |
|------------------------|-----------------------|--|
| No argument | X ↵ | Perform operation X |
| 1 argument | nX ↵ | Perform operation X on the referenced line. |
| 2 argument | m,nX ↵ | Perform operation X on lines m through n, inclusive. |
| 3 argument | m,n\$pX ↵ | Used by MOVE Command only. |

The arguments m and n, which refer to numbered lines in the buffer must be positive and n must be equal to or greater than m.

The three argument "MOVE" command requires a dollar sign between the second and third arguments. "p" may be any line within the text buffer.

Two arguments must be separated by a comma but no comma is allowed between the arguments and the command. Note also, that in order to be executed, a command must be followed by a carriage return (↵).

4.2 SPECIAL CHARACTERS AND FUNCTIONS

A number of keys have special operating functions. These keys and their associated functions are listed below (the non-printing characters are noted; the symbols for these are shown in parentheses, all others echo the character in parenthesis).

4.2.1 CARRIAGE RETURN (↵), non-printing)

Striking the carriage return key, in both Command and Text Mode, signals the Editor to process the information just typed. In Command Mode, it allows the Editor to execute the command just typed. A command will not be executed until it is terminated by striking the carriage return key. In Text Mode, it causes the line of text which it follows to be entered into the text buffer. A typed line is not actually part of the buffer until terminated by a carriage return.

4.2.2 LEFT ARROW (←)

The left arrow (←) is used for error recoveries in both Command and Text Mode. When used in Text Mode, the left arrow cancels everything to the left of itself back to the beginning of the line. The user then continues typing on the same line. When used in Command Mode, the left arrow cancels the entire command and the Editor issues a question mark and a carriage return/line feed. However, the left arrow cannot cancel past the carriage return in either Command or Text Mode. For example:

```
A (CR/LF) ←  
  
PHIS  
"HERE IS A TEXT MODE EXAMPLE"(CR/LF)
```

Only the part in quotes is entered in the buffer.

The Editor generates a carriage return/line feed to indicate that the entire line of text has been deleted.

4.2.3 RUB-OUT (\ , the reversed slash)

Rub-out is also used in error recovery in both Command and Text Mode. Since the read function is performed under control of the Text Mode, the following rules also apply to the read function (however, (\) is not echoed when a rub-out is encountered in the read function). Typing a rub-out echoes a (\) and deletes the last typed character. Repeated rub-outs delete from right to left back to, but not including, the carriage return separating the current line from the previous one. For example:

```
THE QUICK \ \ \ \ ICK BROWN FOX will be entered in the buffer as:  
  
THE QUICK BROWN FOX
```

If an entire line is deleted with rub-outs, the carriage return/line feed is generated as though a left arrow (←) was used. When used in Command Mode, rub-out is equivalent to left arrow (←) and cancels the entire command.

4.2.4 FORM FEED (CTRL/FORM, non-printing)

Form-feed signals the Editor to return to Command Mode. A form-feed character is generated by depressing and holding the CTRL key and typing the FORM key. This combination is typed while in Text Mode to indicate that the desired text has been entered and that the Editor should now return to Command Mode. The Editor generates a carriage return/line feed in response to a CTRL/FORM to indicate the transition back to Command Mode. However, if the Editor is already in Command Mode when the CTRL/FORM is typed, the carriage return/line feed will be generated and the Editor will remain in Command Mode.

4.2.5 PERIOD (.)

The Editor keeps track of the implicit decimal number of the line on which it is currently operating. At any given time the symbol (.) stands for this number and may be used as an argument to a command. For example:

.L means list the current line.

.-1, .+1L means list the line preceding the current line, the current line and the line following it.

After a Read or Append command, the current line counter (.) is the number of the last line in the buffer. After an insert or change command, (.) is equal to the number of the last line entered. After a list command, (.) is the number of the last listed line. After a delete command, (.) is the number of the line immediately following the deletion. After a Get command, (.) is the number of the line typed by the Get. After a Move command, the current line counter (.) is equal to the line immediately following the last line moved. After a Hunt command, the current line counter (.) is equal to the last line number typed by the Hunt command.

4.2.6 SLASH (/)

The symbol (/) has a value equal to the decimal number of the last line in the buffer. It may also be used as an argument to a command. For example:

10,/L means list from line 10 to the end of the buffer.

4.2.7 LINE FEED (non-printing)

Commands are terminated by carriage return/line feed combinations and the lines of each page of text are separated by a carriage return. The user need only strike the return key, however, to terminate a command or input lines since the Editor automatically generates

the line feed to follow each carriage return. During the read function, line feed characters are completely ignored. On output, the Editor automatically punches a line feed following each carriage return. Typing a line feed in Command Mode is equivalent to typing ".+1L" and will cause the Editor to print the line following the current one and increment the value of the current line pointer (.) by 1. During use of the Search Command, line feed will cause the line to be divided into two lines at the point where line feed was typed.

4.2.8 LEFT ANGLE BRACKET (<)

Typing left angle bracket (<) while in Command Mode is equivalent to typing ".-1L" and will cause the Editor to echo (<) and then type out the line preceding the current line. The value of (.) is decreased by 1 so that it still refers to the line last typed.

4.2.9 COLON (:)

The colon sign is used in conjunction with the pointers period (.) and slash (/). When typed in Command Mode, it causes the Editor to print the decimal value of the argument preceding it, followed by a carriage return/line feed. In this way the number of the current line may be found (.:XXXX) or the total number of lines in the buffer (/:XXXX) or the number of some particular line (/ -8:XXXX) may be determined without counting from the beginning.

4.2.10 BLANK TAPE AND LEADER/TRAILER

Blank tape and line feeds are completely ignored during input. Line feeds are automatically replaced wherever necessary on output. Blank tape is not replaced.

4.2.11 TABULATION (→), non-printing)

The Editor is written in such a way as to simulate "tab stops" at 8 space intervals across the carriage. When the user holds the CTRL key and strikes the TAB (CTRL/I) key, the Editor produces a tabulation. The tabulation consists of one to eight spaces depending on the number needed to bring the carriage to the next tab stop. Thus, the user may use the Editor to produce neat columns on the hard copy. In addition, the tab character is used to save space in the text buffer. Whenever possible, the Editor will insert a tab character into the text buffer rather than the corresponding number of spaces. These tab characters are also generated on output, making output tapes relatively shorter. Spaces, however, replace the tabs in any hard copy list function of the text Editor.

NOTE

Location 1047₈ contains the number of spaces used to simulate "tab stops". To change the tabulation, change the constant in location 1047₈ after loading the Editor.

4.3 ND812 SWITCH REGISTER INPUT/OUTPUT DEVICE SELECT OPTIONS

The Editor uses seven ND812 Front Panel SWITCH REGISTER switches in conjunction with the reading and writing of tapes. Bit 0 controls special formatting features of the ND Text Editor.

NOTE

Bit number refers to ND812 SWITCH REGISTER switch number, "1" refers to switch in Up position, and "0" refers to switch in down position.

If bit 0 is set to "1" and a space is typed as the first character of a line, it is converted to a tab, both on the keyboard and in the text buffer. Further input then will begin in column nine of the line. This feature simplifies the input of an untagged line in a source program which should preferably begin in column nine. If a non-space is entered as the first character of a line, input will continue normally for the rest of the line. If a comma is entered in any columns 1 through 8, the Editor will enter a tab after the comma and further input will continue from column nine. This simplifies the typing of a tagged line in a program source tape, where address tags begin in column 1 and the balance of the statement begins in column 9. Finally, if a three letter mnemonic is entered beginning in column 9 so that a space falls in column 12, the Editor will insert an extra space so that further input will continue in column 14. This vertically aligns the address of 3- and 4-letter mnemonics for a neater appearance. Bit 1 enables parity and bit 2 controls the eight level punches.

| Bit 1 | 2 | Operation |
|-------|---|--|
| 0 | 0 | No eight level punches (no parity). |
| 0 | 1 | Eight level punches on all characters (no parity). |
| 1 | 0 | Even parity. |
| 1 | 1 | Odd parity. |

4.3.1 INPUT DEVICE SELECT BITS 8 AND 9

4.3.1.1 LOW SPEED INPUT (BIT 8 = 0, BIT 9 = 0). The Editor expects symbolic paper tape to be loaded in the low speed paper tape reader. The control switch on the paper tape reader should be in the START position. Input will continue until a form-feed (0214g) is encountered on the paper tape. If a form-feed does not appear on the tape, the Editor will continue reading until the text buffer is full or until it runs out of tape. If the reader runs out of tape, the user should type the CTRL/FORM combination in order to return to Command Mode. When input is complete, the control switch on the low speed reader should be returned to STOP position before a command is entered and further editing is continued.

4.3.1.2 HIGH SPEED INPUT (BIT 8 = 0, BIT 9 = 1). The Editor expects the symbolic paper tape to be loaded in the high speed reader. Input will continue until the text buffer

is full or until a form-feed is read on the input tape or until the reader runs out of tape. This command always returns to Command Mode.

4.3.1.3 CASSETTE TAPE INPUT (BIT 8 = 1, BIT 9 = 1). Symbolic Magnetic Tape Cassette is read from Tape Transport 1 into the text buffer. Input continues until the text buffer is full or until a form-feed is read in the input text stream. When complete, this command returns to Command Mode.

4.3.2 OUTPUT DEVICE SELECT BITS 10 AND 11

4.3.2.1 LOW SPEED PAPER TAPE OUTPUT (BIT 10 = 0, BIT 11 = 0). All punching will be done via the Teletype punch. When the desired text has been punched the user should turn the punch off before continuing with editing. An extra carriage return/line feed is added to the text block.

4.3.2.2 HIGH SPEED PAPER TAPE OUTPUT (BIT 10 = 0, BIT 11 = 1). All punching will be done via the High Speed Paper Tape Punch. When the specified text has been punched, the command will return to Command Mode. The punch automatically turns itself on and off.

4.3.2.3 CASSETTE OUTPUT (BIT 10 = 1, BIT 11 = 1). All output will be to the cassette on Tape Transport 2. The user should be certain that his cassette has been write-enabled before entering this command. When the specified text has been written on tape, the command will return to Command Mode.

4.3.3 KEYBOARD/PRINTER INTERRUPT

All operations which cause output to the teleprinter/punch are subject to operator interrupt via the keyboard. If a character is typed on the keyboard while the Editor is printing, the current operation will be aborted and the typed character processed. Although the treatment of the type character differs depending upon the command being executed, the general rule is that the Editor responds to the typed character as though it had completed the interrupted operation. Specific response to keyboard interrupt is included in the following command descriptions. In the case of those commands which involve output to the teleprinter/punch and therefore respond to keyboard interrupts, it is possible to interrupt the current command at any time after the carriage return was typed by the user. Consequently the carriage return/line feed normally echoed may be suppressed, creating a disconcerting, but harmless, effect on the format of the command. This extremely powerful feature relieves the experienced user from the need to wait for the printing of lengthy lines of text when he is able to identify a line by examining only the first few characters.

4.4 COMMAND DESCRIPTION

Commands to the Editor are grouped under three general headings; Input Commands, Output Commands, and Editing Commands. Each command description will state if the Editor returns to Command Mode after completing the operation specified by the command. The Editor will print an error message consisting of a question mark (?) whenever the user has

requested non-existent information or has used an inconsistent or incorrect format in typing the command. For example, if the command requires two arguments and only one (or none) is provided, the Editor will print (?), issue a carriage return/line feed and ignore the command as typed.

Similarly, if a non-existent command character is typed, the error message (?) will be typed, a carriage-return and line-feed issued, and the command will be ignored. However, if an argument is provided for a command that does not require one, the argument will be ignored and the normal function of the command performed. Example (underlined characters were entered by user):

| <u>Message</u> | <u>Explanation</u> |
|--------------------------------|---|
| <u>L ↵</u> <u>? ↵</u> | The buffer is empty, the user is asking for non-existent information. |
| <u>7,5L ↵</u> <u>? ↵</u> | The arguments are in the wrong order; the Editor cannot list backwards. |
| <u>17\$10M ↵</u> <u>? ↵</u> | This command requires two arguments before the (\$); only one was provided. |

4.4.1 INPUT COMMANDS

Input Commands are used to load the buffer from outside sources other than the keyboard.

R ↵ - Read a block of text.

Depending upon the position of Switch Register Bits 8 and 9, reading will be done from the teletype reader, high speed reader or cassette Tape Transport 1. The Editor will read information from the input tape until a form-feed character (CTRL/FORM key combination) is detected, or until the high speed reader runs out of tape or a "\$" is read. All incoming text, except the form-feeds, line-feeds, blank tape and rub-outs are appended to the contents of the text buffer or inserted before line number argument preceding the Read Command. Rub-outs detected on the tape will cause the deletion of the previous characters; information already in the buffer remains there. If the form-feed does not appear on the tape when input is via the teletype reader, a form-feed must be entered via the keyboard to return the Editor to Command Mode. There is no keyboard interrupt since the teleprinter/punch is not involved.

nR ↵ - Read a block of text, inserting it in front of line n.

A ↵ - Append.

Append the incoming text from the teleprinter keyboard to the information already in the buffer (the buffer may be empty initially). The Editor will enter the Text Mode upon

receiving this command and the operator may then type a number of lines of text on the keyboard. The new text will be appended to the information already in the buffer, if any, until the form-feed (CTRL/FORM) key combination is struck.

By giving the Append Command with an empty buffer, a symbolic program tape may be effectively generated on line by entering the program via the keyboard.

Any rub-out encountered during execution of the Append, Insert, or Change Commands will delete the last character typed. Repeated rub-outs will delete from right to left up to but not beyond the beginning of the current line. Lines not terminated with a carriage return are not entered in the text buffer and are therefore lost.

The only teleprinter/punch output by this command is related to simulation of a tab when either CTRL/I (horizontal tab) or CTRL/K (vertical tab) combination is typed (input from the keyboard at all other times is entered into the text buffer). A character typed while simulation of a tab is in progress, will cause the tab to be stopped (although it will have been entered in the text buffer) and the new character to be treated as though the tab were complete.

4.4.2 OUTPUT COMMANDS

Output Commands do not affect the contents of the text buffer.

L ↻ - LIST the entire page.

This command causes the Editor to list the entire contents of the text buffer, inserting spaces for tabs, and generating carriage-return/line-feeds for each carriage-return encountered in the text buffer. A keyboard interrupt terminates the list command; the character typed is processed in Command Mode. The current line is equal to the line being typed when the interruption occurred.

nL ↻ - nLIST line n.

Line number n will be typed out followed by a carriage-return and a line-feed.

m,nL ↻ - m,nLIST lines m through n inclusive.

Lines m through n will be printed on the teletype. The Editor returns to Command Mode after the List Command and the value of the current line (.) is updated to be equal to the line number of the last line printed.

The following commands control the writing of leader/trailer text and form-feeds onto paper or magnetic tape. Before using the Next or Write commands the user should be sure that the output device specified is write-enabled, and that the ND812 SWITCH REGISTER has been properly set up before typing the final carriage return. The Editor automatically returns to Command Mode after executing any command which writes on paper on paper or magnetic

tape. The Editor is designed to minimize the possibility of illegal or meaningless characters being punched into a source tape. Therefore, all characters produced by the Editor are legal characters for the Assembler. In addition, a checksum for each block is written following each Form-Feed separating each block. These numbers are verified on input and "ERROR" generated if in error. A tape containing illegal characters may be corrected by simply reading it into the Editor and punching it out. Depending upon the position of the ND812 SWITCH REGISTER Bits 10 and 11, output will be either the Teletype, the High Speed Punch, or the Magnetic Tape Cassette tape transport 2.

When output is to the teleprinter/punch, a keyboard interrupt terminates output; the character typed is processed in Command Mode.

W - WRITE the entire contents of the text buffer.

If using the low-speed punch, turn it "ON" before typing the final carriage return.

nW - WRITE line n only.

m,nW - WRITE lines m through n inclusive.

This command requires that m must be less than or equal to n.

P - PUNCH

Stop the Editor to permit turning on punch device. Depressing CONTINUE starts punching of text on selected paper tape device.

F - FORM FEED

This command causes the punching of five blanks, a form-feed character, and approximately two inches of blank tape. If using low speed punch, turn the punch ON before typing the final carriage return terminating the command. Turn punch off after form feed and checksum characters are punched and before continuing editing.

T - TRAILER

This command causes about four inches of blank tape to be written. If using the low speed punch, turn punch on before typing final carriage return. Turn punch off before completion of the command and before continuing editing.

U - REWIND Cassette.

The Input Cassette will be rewound if Bits 8,9 = 1. The Output Cassette will be rewound if Bits 10,11 = 1. Formatting for cassette output is also re-initialized (see Open Command). If a cassette drive is selected that contains no cassette the Editor will not return. Depress ND812 STOP key, set SWITCH REGISTER to address 0200, and depress CONTINUE.

B - BACKSPACE Cassette

Backspace cassette to next filemark (or BOT).

O - OPEN FILE on Output Cassette

This command must be given before writing the first block of a new source tape. This command clears the checksum and tells the Editor to begin dividing the output into fixed-length records of 1024 characters each. This does not affect the user blocking and, in fact, is not necessary for use of the Editor. Improperly formatted cassette tapes may be read without error by the Editor. The Assembler, however, has a limited input buffer capacity and can read blocks of no more than 1024 characters in length. The extra inter-record gaps inserted on the cassette during output from the Editor are otherwise ignored by the Editor and Assembler. The "U" Command includes an effective Open File if a cassette unit was selected as the output device.

N - NEXT

This is a utility command which combines the functions of three commands. It writes the contents of the buffer on the device specified by ND812 SWITCH REGISTER Bits 10 and 11, writes blank tape, a form feed, checksum, more blank tape, clears the buffer, and reads in the next page of text from the device specified by ND812 SWITCH REGISTER Bits 8 and 9.

nN - Executes the above sequence n times.

If n is greater than the number of blocks of input tape, the command will proceed in the specified sequence until it reads the end of the input tape. Then it will return to Command Mode. The command will also terminate by typing a (?) upon reading two consecutive form-feeds. When using the low-speed reader, it may also be terminated by setting the reader control switch to the "STOP" position and typing two consecutive Form-Feeds (CTRL/L). When using the low-speed punch, keyboard interrupt will terminate the command and cause the character typed to be processed in Command Mode.

4.4.3 EDITING COMMANDS

The following commands permit deletion, alteration or expansion of text in the buffer.

nD ↻ - DELETE line n.

Line n is removed from the text buffer. The numbers of all succeeding lines are reduced by 1. The last line number is decremented.

m,nD ↻ - DELETE m through n inclusive.

The line following n becomes the new line m and the rest of the lines are renumbered

accordingly. The value of the current line counter, (.), will be equal to "m". The Editor returns to Command Mode after all delete operations.

nI ↻ - INSERT the typed text before line n.

The typed text is entered before line n until a form-feed (CTRL/FORM) is encountered.

The Editor enters Text Mode to accept input and the first line typed becomes the new line n. Rub-outs are recognized. Both the total line count and the number of all lines following the insertion are increased by the number of lines inserted. The value of (.) is equal to the number of the last line inserted. To re-enter the Command Mode, the form-feed key combination must be entered to terminate Text Mode. If this is not done, all subsequent commands will be interpreted erroneously as text and entered in the program immediately after the insertion.

Keyboard interrupt affects tabs as described under the Append Command above.

I ↻ - INSERT without an argument will insert text before line 1.

nC ↻ - CHANGE line n.

Line n is deleted and the Editor enters Text Mode to accept input. The user may now type as many lines of text as he desires in place of the deleted line. If more than one line is inserted, all subsequent lines will be automatically renumbered and a line count updated appropriately.

m,nC ↻ - CHANGE lines m through n inclusive.

This command requires that m must be numerically less than n.

Lines m through n are deleted and the Editor enters Text Mode allowing the user to type in any number of lines in their place. All subsequent lines will be automatically renumbered to account for the change and the line count will be updated. After any change operation, return to Command Mode is accomplished by entering a form-feed (CTRL/FORM key combination) to terminate input. After a Change, the value of the current line number, (.), is equal to the number of the last line of the change. Lines which are changed or deleted physically disappear from the buffer area, thus the amount of buffer area used is always directly related to the current size of the symbolic text.

Keyboard interrupt affects tabs as described under the Append Command.

m,n\$KM ↻ - MOVE lines m through n inclusive to before k.

This command requires that m must be numerically less than n; and k may not be in the range between m and n.

Lines m through n are moved from their current position and inserted before line k. The lines are renumbered after the move is completed although the value of the current line pointer, (.), is unchanged. Moving lines require additional buffer space equal to the number of lines moved. The Move Command must have three arguments. However, arguments m and n may be equal allowing the user to move a single line. The Editor returns to Command Mode upon completion of the Move Command.

G ↻ - GET and list the next line starting with a symbolic tag.

Beginning with the line following the current line, the Editor searches the text buffer for a line containing a symbolic tag, types the line on the teletype and returns to Command Mode. The current line pointer, (.), is updated to the typed line.

Keyboard interrupt terminates the command and causes the character typed to be processed in Command Mode.

nG ↻ - GET and list first line after n with a symbolic tag.

The Editor begins with line n and tests it and each succeeding line as described above. Both G and nG update the current line counter, (.), after finding the specified line. However, if either version of the Get Command reaches the end of the buffer before finding a line with a comma, the current line counter remains unchanged and a (?) is typed to indicate that no line was found. The Editor returns to Command Mode after completion of the Get Command.

nS ↻ - SEARCH line n for the character specified.

After the carriage-return, the Editor searches line n for the specified character, allowing modification of line when character is found.

The Search Command is one of the most useful functions in the Editor. It is also structured somewhat differently from the other Editor Commands. After terminating the command nS with a carriage-return, the user has told the Editor to search line n but he hasn't specified what to search for.

The Editor is therefore, waiting for the user to type a character. The character he types is taken as the object of the search but is not echoed. The Editor instead immediately begins typing out the specified line. After typing the character for which it is searching, the Editor stops. All the text editing features are then available to the user. He may proceed by using any of the following:

- a. Delete the entire typed portion of the line by typing the left arrow (←).
- b. Delete the entire untyped portion and terminate the line and the search by typing carriage-return (↻).
- c. Delete from right to left one of the typed characters for each rub-out typed.

d. Insert characters after the last one typed by simply typing them.

e. Return to initial status of the Search Command by typing CTRL/G. The user may now enter a different search character, a line-feed which will insert a carriage-return in the text buffer dividing the line into two or a CTRL/FORM which will cause the Editor to search the line for the old character specified. When typing stops, all options are again available.

Keyboard interrupt terminates the command; the character typed is processed in Command Mode.

H ↻ - HUNT through the text buffer for specified character string.

After terminating the command with a carriage-return, the user has told the Editor to search for a character string but has not specified the character string. The Editor is then waiting for the user to specify a character string. All the special characters such as rub-out, left arrow (←), are available to the user in forming the character string. When the character string is complete, the user terminates it with a carriage-return and the Editor proceeds to search from the beginning to the end of the text buffer for the appearance of the string in any line. The Editor lists the line number and the text of each line in which the character string is found. The current line pointer, (.), is set equal to the last line listed.

Keyboard interrupt terminates the command. The character typed is processed in Command Mode. The current line pointer (.) is set equal to the line number of the line being typed when the interruption occurred.

E ↻ - ENTERED characters.

Types number of characters currently in the buffer (maximum 4095 characters).

K ↻ - KILL contents of the text buffer.

The last line number is set to zero. The contents of the buffer area is not actually altered, except when any text modification command is entered. The "Y" command may be used to restore the buffer.

Y ↻ - Y RESTORE the contents of the buffer after a KILL command.

If the text buffer is mistakenly "killed", it may be restored with this command. The "Y" command will echo a question mark "?" if the text buffer has been modified in any way since the last KILL command was entered. Any modification command entered following a KILL will prevent the restoration of the buffer contents.

nX ↻ - MOVE specified TAG to line n.

After the carriage-return/line-feed is echoed, the user may enter a character string of

up to six characters, followed by a carriage-return. The Editor will search the buffer for the character string followed by a comma, and, when found, delete them from the line on which they appear. Then, the command will add the character string, including the appended comma, to the beginning of line n.

This command is extremely useful for moving symbolic tags to statements from another line without the need to retype or modify those two lines of text.

If the specified tag cannot be found, the Editor will type a question mark "?" and return to Command Mode.

m,nZ ↻ - ALPHABETIZE lines m through n.

Line m must be less than or equal to n. Lines m through n will be arranged in alphabetical order according to ASCII coding pivoted for numerals and special characters. Thus, "A" has minimum value, and numbers appear after letters. A "tab" is ignored which is an exception to the coded weights. Arguments must be provided.

This command is useful for alphabetizing long lists of definitions or sorting lists of numbers.

SECTION V COMMAND SUMMARY

5.1 INTRODUCTION

This section summarizes all of the commands and special characters which control the functions of the Editor. Those commands which require or allow arguments, are listed with the maximum number of arguments as represented by lower case letters.

5.2 SPECIAL CHARACTERS

Carriage Return () - Terminates commands and lines (Command and Text Mode).

Left Arrow () - Deletes the current line or command (Command and Text Mode).

Rub-Out () - Delete the previously typed character (Text Mode).

Form-Feed - Returns program to Command Mode (Text Mode).

Period (.) - Represents the current line number.

Line Feed (non-printing) - Causes printing of the next sequential line of the buffer. Increments current line number (Command Mode). The line feed is ignored in all Text Mode operations except in Search, where line feed can be used to divide a line into two or more parts (Text Mode).

Left Angle Bracket (<) - Causes printing of previous line of text buffer. Decrements current line number (Command Mode).

Colon (:) - Causes printing of the decimal value of the argument preceeding it. Generally, used to determine the actual value of (.) or (/) (Command Mode).

Tabulation (CTRL/I, non-printing) - Causes 8 space "tab" of carriage (Text Mode).

5.3 EDIT COMMANDS

R-----READ

Read a block of text (Command Mode).

A-----APPEND

Append text from keyboard to text buffer (Command Mode).

m,nL---LIST

List lines of text buffer on teleprinter (Command Mode).

m,nW---WRITE

Write all or part of text buffer (Command Mode).

m,nP---PUNCH

Punch all or part of text buffer. Stops to turn on device - press CONTINUE.

F-----FORM-FEED

Generates blank tape and a form-feed on specified output device to separate blocks of text (Command Mode).

T-----TRAILER

Generates about 4 inches of blank tape (Command Mode).

U-----REWIND

Rewinds magnetic cassette tapes.

B-----BACKSPACE

Backspaces cassette to previous filemark (or BOT).

O-----OPEN FILE

Open file on output cassette, resets character counter to 1024 and clears writing checksum.

nN-----NEXT

Writes the current block, generates a form-feed, clears the buffer and reads the next block of text (Command Mode).

m,nD---DELETE

Delete all or a portion of text (Command Mode).

nI-----INSERT

Insert text entered from keyboard in front of line "n" (Command Mode).

m,nC---CHANGE

Change all or a portion of the text to text entered from keyboard (Command Mode).

m,n\$kM-MOVE

Move a portion of text to another part of text buffer. Must have three arguments, but m may be equal to n (Command Mode).

nG-----GET

Get the next line in the text buffer starting with a symbolic tag (Command Mode).

nS-----SEARCH

Search line for a specified character. Allows modification of line with normal text modification functions (Command Mode).

H-----HUNT

Hunt through text buffer for specified character string. Lists lines containing the string with their line numbers (Command Mode).

E-----ENTERED CHARACTERS

Causes number of characters entered in buffer to be typed. The maximum number of characters which may be entered in the buffer is 4095 (Command Mode).

K-----KILL

Kills the contents of the text buffer.

Y-----RESTORE

Restores the contents of the text buffer after a KILL command was entered.

nX-----MOVE TAG

Moves specified TAG to line n.

m,nZ---ALPHABETIZE

Alphabetizes lines m through n.

!-----BUFFER OVERFLOW

The exclamation point indicates that an operation was attempted which required more buffer space than was available (Text or Input Mode).

APPENDIX A

LOADING AND INITIALIZATION OF THE ND812 TEXT EDITOR

The ND Text Editor (ND41-0002) may be loaded into the ND812 Computer via Teletype or optional High Speed Reader, Low Speed Reader, or Tape Cassette. The following is a step-by-step procedure for loading and initialization of the ND812 Text Editor using the above devices.

LOADING AND INITIALIZATION USING TELETYPE

- a. Depress ND812 Computer STOP switch.
- b. Set Teletype START/FREE/STOP switch in FREE position.
- c. Place the ND812 Text Editor program into the Teletype Reader with the leader (level 8 punched) over read head.
- d. Set Teletype START/FREE/STOP switch to START position.
- e. Simultaneously depress ND812 LOAD AR and NEXT WORD switches. The Teletype Reader should step through the paper tape leader and read the program into the ND812 Memory. Upon completion of tape read-in, the Reader automatically stops. After the Reader stops, set ND812 Computer SELECTED REGISTER switch in J position, and verify that J-register is zero (all lamps off). If J-register is not zero, repeat steps a through e.

LOADING AND INITIALIZATION USING HIGH SPEED READER, LOW SPEED READER, OR TAPE CASSETTE

The ND812 Text Editor is loaded into the ND812 Computer via the Binary Loader Program (ND41-0005) as follows.

- a. Place Binary Loader Program tape leader (level 8 punched) over reader head.
- b. Set Teletype START/FREE/STOP switch in START position.
- c. Simultaneously depress ND812 LOAD AR and NEXT WORD switches. The

hardware loader then loads the Binary Loader. Tape motion stops automatically when the tape trailer (level 8 only punch) is sensed. Set ND812 Computer SELECTED REGISTER switch in J position, and verify J-register is zero (all lamps off). If J-register is not zero, repeat steps a through c.

- d. Set ND812 SWITCH REGISTER switches to $77\emptyset\emptyset_8$.
- e. Depress ND812 LOAD AR switch.
- f. If High Speed Reader is used, set SWITCH REGISTER switches to $37\emptyset\emptyset_8$.
- g. If low speed reader is being used, leave SWITCH REGISTER switches at $77\emptyset\emptyset_8$.
- h. Place ND812 Text Editor tape leader (level 8 punched) over reader head.
- i. If a tape cassette is being used, set SWITCH REGISTER switches \emptyset and 1 to " \emptyset ". Set SWITCH REGISTER switches 2, 3, and 4 to the cassette number desired (cassette #1 = $\emptyset\emptyset 1$, cassette #2 = $\emptyset 1\emptyset$, cassette #3 = $1\emptyset\emptyset$). Since more than one program may be on a cassette, switches 5 through 11 are used to indicate the tagword to be selected. This limits tagwords to $\emptyset\emptyset\emptyset\emptyset_8$ through $\emptyset 177_8$.
- j. Depress ND812 START key.

The Binary Loader loads the ND812 Text Editor. When the tape stops, set ND812 SELECTED REGISTER switch in K position and verify that the K register is zero. If not, repeat steps d through j.