

Aalborg Universitetscenter
Institut for Elektroniske Systemer
Laboratoriet
Badehusevej 1A
9000 Aalborg

001425

ID-7000 TEXTEDITOR

for the
ID-7000 MICROPROCESSOR SYSTEM
USERS MANUAL
July 1976

Written by Steen Andreassen

C O N T E N T S

SECTION 1	OPERATING FEATURES	
1.1	INTRODUCTION	3
1.2	ENTERING THE TEXTEDITOR	3
1.3	FORMAT OF SOURCE TEXT	3
1.4	FORMAT OF COMMANDS	4
1.5	LINE COUNT CHARACTERS	4
1.6	INPUT/OUTPUT UNITS	5
1.7	SIZE OF TEXTBUFFER	6
1.8	ERROR REACTION	7
SECTION 2	COMMAND REPERTOIRE	
2.1	INPUT COMMANDS: READ, APPEND, BEGIN	8
2.2	OUTPUT COMMANDS: LIST, PUNCH, FORM- FEED, TRAILER, NEXT	9
2.3	EDITING COMMANDS: KILL, DELETE, INSERT, CHANGE, MOVE, SEARCH	11
SECTION 3	SUMMARY OF OPERATION	
3.1	SPECIAL KEY FUNCTIONS IN COMMANDS	15
3.2	SPECIAL KEY FUNCTIONS IN TEXT	16
3.3	SUMMARY OF COMMANDS	17

SECTION 1. OPERATING FEATURES

1.1 INTRODUCTION

The texteditor reads the source text from paper tape into a textbuffer. By means of keyboard commands the text may be corrected, parts of the text may be deleted and new text can be inserted by typing it on the console. After the necessary corrections the text is punched on paper tape.

1.2 ENTERING THE TEXTEDITOR

The normal entry point of the texteditor is X.D000. The texteditor can however also be entered at X.D003 as described in the sections 1.6 INPUT/OUTPUT UNITS and 1.7 SIZE OF TEXTBUFFER. If the texteditor is entered from the ID-7000 DEBUGGER this may be achieved by the command D000<OXJ. The texteditor will answer by typing EDIT and an underscore _ on the next line. This means that the texteditor is waiting for a command to be typed on the console device. Whenever the texteditor is returning to command mode, i.e. waiting for the next command, this is signalled to the operator by the underscore.

1.3 FORMAT OF SOURCE TEXT

The source text is organized in pages and lines. The pages are separated by formfeed characters (FF) the lines by the sequence: LF, CR and five NULL characters. The LF and the NULL characters are blind characters during input. The texteditor punches these characters on the output tape to make the format conform with the input format of the ID-7000ASSEMBLER. Neither pages nor lines have to be written in a fixed format. When reading text from paper tape the texteditor reads one page at a time and the total number of characters on a page should therefore be less than the size of the textbuffer. The standard size of the textbuffer is approximately 2000 characters allowing more than 60 lines with an average of 30 characters. The parity of the input characters is ignored. All printing characters (i.e. all characters with ASCII values greater than or equal to 32) are legal characters.

Small letters will not be converted to capital letters. Most of the control characters and the delete (DEL) character are blind. Exceptions are BEL (control/G), TAB (control/I), FF (control/L) and CR. BEL and FF are equivalent and function as page separation characters. CR is the line separation character. A TAB (tabulation) character in the input text will be interpreted as a number of spaces sufficient to bring the line cursor to the next column, which is a multiple of 8. The texteditor thus has software tabulations in character positions 8, 16, 24, ...

1.4 FORMAT OF COMMANDS

The reading, correction and punching is controlled by commands typed in from the console. Examples of commands are:

R↓	Read one page
5D↓	Delete line 5 in the text buffer
1,)-1P↓	Punch all lines except the last
F↓	Punch a Formfeed to separate pages in output

All command are preceded by from zero to three parameters. After typing the parameters and the single letter designating the command, execution of the command is started by typing CR. The parameters must either be an integer number N or an expression containing the special line count characters.

1.5 LINE COUNT CHARACTERS

By convention the source text, i.e. the text in which corrections are to be made, is organized in pages and lines. When the first page of the source text is read into the textbuffer the lines within the page is implicitly given consecutive numbers: The first line no. 1, the second line no. 2 etc. Whenever lines are deleted, inserted or moved this implicit numbering is updated. Three line count characters (, . and) are available. (is equal to the number of the first line in the textbuffer,) to the last line and . is the current line pointer. E.g. after the execution of a LIST command . is equal to the number of the last line listed.

(, . and) can be used when writing the command para-

meters. A line count character, e.g. `.` may be used in the following combinations:

- 1) `.`
- 2) `. + N`
- 3) `. - N`

where N is a decimal number. Thus `)-1L` is a valid command to list the last line but one, while `1+.L` is invalid.

The values of the line count character may be inspected by typing the character followed by `:` (colon) or `=` (equal).

The texteditor responds by typing the value.

Example:

```
( = 00036
```

When the first page is read from the papertape reader into the textbuffer, the value of `(` is equal to 1. The execution of a KILL command causes the line count character `(` to be incremented by the value of LINESREAD. LINESREAD is the number of lines input from the paper tape reader since the last execution of a KILL command. The KILL command also resets the value of LINESREAD to zero. Note that a KILL command is implicitly executed as part of a NEXT command. This system will permit the operator to use the line numbers from the source text during editing. Every time a new page is read, there will be correspondance between the line numbers used by the text-editor and the linenumbers in the source text. If it during editing is desired to change the value of `(` it may be done by the BEGIN command. Executing the command `nB` will assign the value `n` to `(`.

1.6 INPUT/OUTPUT UNITS

The texteditor operates with 4 logical input-output devices.

- 1) Console device. Used by the operator for writing the editing commands and for writing new text to be appended to or inserted into the text buffer.
- 2) Reader for paper tape input. Contains the text to be corrected.
- 3) Punch for paper tape output. The text is output on this unit after correction.
- 4) List device. Used by the LIST-command for listing the complete text or parts of it.

The assignment between physical and logical devices is determined by the value of the I/O status byte (byte 0). The I/O status byte has the following format.

7	6	5	4	3	2	1	0
LIST FIELD:		PUNCH FIELD:		READER FIELD:		CONSOLE FIELD:	
00 TTY		00 TTY		00 TTY		00 TTY	
01 DUMMY		01 PTP		01 PTR		01 DUMMY	
10 DUMMY		10 DUMMY		10 DUMMY		10 BATCH	
11 USER L		11 USER P		11 USER R		11 USER C	

(For further information see ID-7000 Utility Routines, DDE, Juni 1976).

When entering the texteditor at the normal entry point X.0000 the I/O status byte is initialized to a standard value suitable for the configuration of physical I/O devices available at the installation. This standard I/O configuration is chosen at the time of installation of the texteditor. If the user wants another assignment between physical and logical devices this may be done by initializing the I/O status byte to the desired value before entering the texteditor. When the texteditor is entered at the entry point XD 003 the value of the I/O status byte is left unchanged.

1.7 SIZE OF TEXTBUFFER

The texteditor uses locations 16 (X.10) to 69 (X.111) as work area. The standard location of the textbuffer is from BB=256 (X.100) to the end of the available RAM-memory. With e.g. 2K RAM-memory this will be BE=2047 (X.7FF). BB and BE are initialized to their standard values when entering the texteditor at X.0000. The textbuffer may be expanded or moved by initializing BB and BE to other values before entering the texteditor and using the entry point X.0003. This will leave BB and BE unchanged. The memory locations for BB and BE are:

64	X.40	BB
66	X.42	BE

Assuming the above mentioned values of BB = 112 and BE = 2047 the textbuffer can contain 1836 characters. This allows for more than 60 lines with an average of 30 characters. For economy consecutive spaces are packed internally as tabulation. A lineshift takes three memory locations. The cost of storing a line in the textbuffer is the number of characters plus three. An empty line occupies three locations. It should be noted that all editing commands except MOVE uses space in the textbuffer.

1.8 ERROR REACTION

The texteditor will print a question mark whenever an illegal command is written. The syntax of the command must conform with the syntax given in the section COMMAND REPERTOIRE. Both the syntax of the command and the range of the parameters is checked before the execution of the command begins. Examples of illegal commands are listed below.

Examples:

<u>Message</u>	<u>Explanation</u>
L?	The buffer is empty. The operator is asking for nonexistent information.
8,5L?	Arguments in wrong order. The texteditor can not list backwards.
4R?	No argument is required with the READ command.
H?	Nonexistent command letter.

SECTION 2

COMMAND REPERTOIRE

2.1 INPUT COMMANDS

R↓ READ

Read one page of text from the paper tape reader. The editor reads information from the input tape until a form feed character (CTRL/L) is encountered. If input is taken from a photoelectric reader end-of-tape will generate a form feed character. The text read in is appended to the text already in the textbuffer. Any RUBOUT encountered during a READ command will be ignored. The number of lines read by the READ command is accumulated in the variable LINESREAD. Note that the READ command is executed implicitly, when a NEXT command is executed.

A↓ APPEND

During the APPEND command the text entered from the console device is appended to the text already in the textbuffer until a form feed (CTRL/L) is met. The APPEND command is used to write new programs or for appending new code to existing programs.

Typing a RUBOUT character will produce a BACKSLASH \ and delete the last character typed. Characters can be deleted from right to left until but not beyond the beginning of the current line.

Typing a BACKARROW ← (CTRL/O) will delete the current line. Typing continues on the next line.

The APPEND command should not be used to read paper tapes as every RUBOUT on the paper tape will delete a character.

APPEND is equivalent to the command)+11↓

nB↓ BEGIN

The value of (is set equal to n. The values of . and) and the line numbers of the lines in the textbuffer are adjusted accordingly.

B↓ (is assigned the value one. Equivalent to 1B↓

2.2 OUTPUT COMMANDS

`n,mL` LIST

The lines `n` through `m` are listed, `n` and `m` inclusive. At the beginning of every line the line number is listed. At any time during the execution of the LIST command, the listing may be interrupted by pressing the BREAK-key. The value of the current line counter is updated to the value of the last line listed.

`nL` Only line `n` is listed. Equivalent to `n,nL`.

`L` The complete textbuffer is listed. Equivalent to `(,)L`.

LF LINEFEED

This option is included for ease of operation of the LIST command. Pressing the LF key will cause the line after the current line to be listed. The value of `.` is then incremented by one. Equivalent to `+.lL`.

ESC ESCAPE

Equivalent to LINEFEED. Note that listing by means of LINEFEED will produce an output with double spacing between lines. ESC only gives single spacing.

List the preceding line. The line before current line is listed. The value of `.` is decremented by one. Equivalent to `-.lL`.

`n,mP` PUNCH

The lines `m` through `n` are punched, `n` and `m` inclusive. If the punch medium is the teletype punch the editor waits for a space to be typed before starting the punching in order to allow the operator to turn on the teletype punch. Thus after typing the PUNCH command and the CR that normally starts execution of a command the operator should switch on the teletype punch and then type a space to initiate punching. After punching the editor again waits for a space to allow the operator to turn off the punch. If the punch medium is not the

teletype punch but e.g. a fast punch the texteditor will not wait. The punching may be interrupted by pressing the BREAK-key. After execution of the punch command the current line counter is updated to the value of the last line punched.

nP↓ Line n is punched. Equivalent to n,nP↓

P↓ The entire page in the textbuffer is output on the punch device. Equivalent to (,)P↓

F↓ FORMFEED

The FORMFEED command causes punching of six blanks, a formfeed character and approximately 25 cm of blank tape. If using the teletype punch the texteditor waits for a space before and after execution of the command as described with the PUNCH command.

T↓ TRAILER

The TRAILER command causes punching of approximately 25 cm of blank tape. If using the teletype punch the texteditor waits for a space before and after execution of the command as described with the PUNCH command.

N↓ NEXT

The NEXT command combines the functions of four command. It punches the contents of the buffer, punches some blank tape, a formfeed and more blank tape, kills the buffer, and reads in the next page of text from the reader. It is equivalent to the sequence P↓ F↓ K↓ R↓. If using the teletype punch the texteditor waits for a space before and after punching.

nN↓ The NEXT command is executed repeatedly until the page containing line n in the source text is read into the textbuffer. If n is less than or equal to (+LINESREAD the command has no effect. The texteditor responds with a ? . For information about the LINESREAD variable see

the sections LINE COUNT CHARACTERS, KILL and READ. If using the teletype punch the texteditor waits for a space from the keyboard only before the first and after the last page is punched. While punching, the execution may be stopped by pressing the BREAK-key.

2.3 EDITING COMMANDS

The following commands are used for making the desired changes, deletions or expansions of the contents of the textbuffer.

K↓ KILL

The entire textbuffer is deleted. The value of the special character (is increased by the value of LINESREAD. LINESREAD is reset to zero.

n,mD↓ DELETE

Lines n through m inclusive are deleted. 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 . is equal to the number of the line after the deleted lines. The value of the line count character) is reduced by m-n+1.

nD↓ Delete line n. Equivalent to n,nD↓

nI↓ INSERT

Textlines typed on the console device is inserted before line n, until a formfeed (CTRL/L) is typed. The first line typed becomes the new line n. The line count and the lines following the insertion are increased with the number of lines inserted. At exit the current line counter points to the last line inserted. Lines are inserted into the textbuffer, when the CR character is typed. (The texteditor automatically supplies the LF character). Thus the last lines typed should be finished by a CR, before typing CTRL/L. Otherwise the last line typed in will not be included in the textbuffer.

NOTE: The texteditor remains in the INSERT command until a CTRL/L is typed.

- I↓ Insert at the beginning of the textbuffer.
Equivalent to: lI↓
- m,nC↓ CHANGE
CHANGE lines m through n, inclusive. Lines m through n are deleted and new lines typed in to replace them.
Equivalent to the sequence: m,nD↓ mI↓.
- nC↓ Change line n. Equivalent to n,nC↓
- m,n,kM↓ MOVE
MOVE lines m through n inclusive to before line k. k should not be in the range between m and n+1. The lines in the textbuffer are renumbered after the move is completed. The current line pointer is set to the first line moved. One or more lines may be moved to the end of the textbuffer by specifying k equal to)+1.
Example: 5,10,)+1M).
Moving lines does not use additional bufferspace.
- m,nS↓ SEARCH
Lines m through n inclusive are searched. The SEARCH command is used for modifying the lines in the textbuffer without completely rewriting the lines. After typing the command and the CR to start execution of the command the texteditor waits for a search character to be typed on the console device. The status is identical to the status described in option g after typing BELL (CTRL/G). When a key is pressed on the keyboard the character is not echoed, but the texteditor starts typing from the beginning of line m. The typing stops, when the search character is encountered. All the editing options are then available to the user.

OPTIONS

- a. Insert characters after the last one typed simply by typing them.
- b. Each RUBOUT typed will delete from right to left one of the typed characters. (As described with the APPEND command).
- c. Delete the entire typed portion of the line by typing BACKARROW ← .
- d. Delete the portion of the line not yet typed by typing CR. The search continues on the next line with all options available.
- e. Insert a CR and a LF, thus dividing the line into two by typing LF.
- f. Continue the search until the next occurrence of the search character by typing FORMFEED (CTRL/L). When typing stops all options are again available.
- g. Change the search character and continue searching by typing BELL (CTRL/G) followed by the new search character. Typing CTRL/G twice will cause the search to proceed to termination, since the search character will now be BELL which is not stored in the buffer. Using CR as search character has the same effect. The use of SPACE and TAB (CTRL/I) as search characters should be avoided. It will give unpredictable results due to the internal representation of spaces as tabulations.

Except when the texteditor is typing in search for the next occurrence of the search character, all the options are available at any point of time.

The execution of the search command stops either when the search reaches the end of line n or if the BREAK-

key is pressed while the texteditor searches for the next occurrence of the search character. In the last case the search is terminated, when the end of the current line is reached.

- nS↓ Only line n is searched. Equivalent to n,nS↓.
- S↓ The whole textbuffer is searched. Equivalent to (,)S↓.
- E↓ Exit. Control is transferred to the DEBUG/MONITOR by the execution of a CALL F008 instruction.

SECTION 3.

SUMMARY OF OPERATIONS

3.1 SPECIAL KEY FUNCTIONS IN COMMANDS

↵	CR (RETURN-key)	Execute the command.
.	Period	Current line counter.
(Left parenthesis	First line in textbuffer.
)	Right parenthesis	Last line in textbuffer. . , (and) can be used as parameters. Ex: .-2,)L).
=	Equal sign	Used in conjunction with . , (and) to obtain their value. Ex: . = 00052.
:	Colon	Same function as = .
LF	LF (LINEFEED)	List the next line. Equivalent to .+1L).
ESC	ESCAPE	List the next line. Equivalent to .+1L).
<	Left angle bracket	List the previous line. Equivalent to .-1L).
,	Comma	Parameter separation character.

3.2 SPECIAL KEY FUNCTIONS IN TEXT

↓	CR (Return key)	Execute the command.
←	BACKARROW (CTRL/O)	Cancel the entire line of text.
	RUBOUT	Delete from right to left one character for each RUBOUT typed. Does not delete past the beginning of the line.
FF	FORMFEED (CTRL/L)	End of text input. Page separation. During SEARCH: Continue searching until the next occurrence of the search character.
BELL	(CTRL/G)	End of text input. Page separation. During SEARCH: Take the next character typed as new search character.
LF	LINEFEED	Blind character. During SEARCH: A CR/LF combination is inserted into the line being searched.
TAB	(CTRL/I)	Interpreted as a number of spaces sufficient to bring the line cursor to the next position which is a multiplum of 8.

3.3 SUMMARY OF COMMANDS

Command	Format(s)	Meaning
READ	R↓	Read text from reader and append to buffer until a formfeed is encountered.
APPEND	A↓	Append incoming text from keyboard to any already in buffer until a formfeed is encountered.
BEGIN	B↓	The linenumber of the first line in the buffer is set equal to 1
	nB↓	The linenumber of the first line in the buffer is set equal to <u>n</u> .
LIST	L↓	List the entire buffer.
	nL↓	List line <u>n</u> .
	m,nL↓	List lines <u>m</u> through <u>n</u> inclusive.
PUNCH	P↓	Punch the entire buffer. If using the TTY-punch, strike the space-key before and after punching.
	nP↓	Punch line <u>n</u> .
	m,nP↓	Punch lines <u>m</u> through <u>n</u> inclusive.
FORMFEED	F↓	Punch trailer, punch a FF, LF and CR , punch trailer. If using the TTY-punch, strike the space-key before and after punching.
TRAILER	T↓	Punch four inches of trailer. If using the TTY-punch, strike the space-key before and after punching.
NEXT	N↓	Punch the entire buffer and a formfeed, Kill the buffer and Read the next page. If using the TTY-punch, strike the space-key before and after punching.
	nN↓	Repeat the above sequence until the page containing line <u>n</u> is read.
KILL	K↓	Kill the buffer.
DELETE	nD↓	Delete line <u>n</u> of the text.
	m,nD↓	Delete lines <u>m</u> through <u>n</u> inclusive.
INSERT	I↓	Insert before line 1 all the text from the keyboard until a formfeed is entered.
	nI↓	Insert before line <u>n</u> until a formfeed is entered.
CHANGE	nC↓	Delete line <u>n</u> , replace it with any number of lines from the keyboard until a formfeed is entered.
	m,nC↓	Delete lines <u>m</u> through <u>n</u> , replace from keyboard as above until formfeed is entered.
MOVE	m,n,kM↓	Move lines <u>m</u> through <u>n</u> inclusive to before line <u>k</u> .
SEARCH	S↓	Search the entire buffer for the character specified (but not echoed) after the carriage return. Allow modification when found.
	nS↓	Search line <u>n</u> , as above, allow modification.
	m,nS↓	Search lines <u>m</u> through <u>n</u> inclusive, allow modification.
EXIT	E↓	Control is transferred to the DEBUG/MONITOR.