

**WANG**

LABORATORIES, INC.

## Virtual Assembly Language Editor

Program description

September 10 1980

revised May 11 1981

The following describes the function, operation and use of the Virtual Assembly Language Editor. This program was written by Max Blomme, 2200 Micro-code development (dept. 40).

## A Purpose

To create and edit assembly language format data files for use as input to various assemblers operating on the 2200.

## B Requirements

28K memory on a 2200VP or MVP (non-syntax checking) or

38K memory on a 2200VP or MVP (syntax checking)

Disk unit (a hard disk is preferred but a floppy may be used)

2226B (80x24), 2236D, 2236DE or 2236DW (last two preferred)

Printer is optional

## C Features and Limitations

The virtual editor must have access to 503 sectors of catalog area on the disk for each user's work file. The edit file can be no longer than 999 lines. Caution must be exercised on long files, for some assemblers allow only up to 511 lines. For source of more than 999 (511) lines, multiple files, with unique file names, must be used. The convention to be used for file naming is that the first two characters are initials and the other six are neither all blank nor all numeric. When the file name is displayed on the CRT or entered from the keyboard, a period is included to separate the initials from the file name (this period is not actually written on the disk, so a LIST DC T will not show it ).

The character set within a line includes all ASCII codes between Hex(10) and Hex(7A) that are available on a keyboard. Care should be taken not to create data which the assembler cannot handle. Backspace, space, and carriage return are used as special characters.

All or part of one or more edit files may be loaded into memory up to the restriction of 999 lines, individual lines may be edited, and groups of lines may be inserted or deleted from the file currently in the work area. The file currently in the work area may be saved, either over an old disk file, or in a newly defined disk file.

Finally, as an option, syntax checking for specific assembly languages is supported (currently Z80, 8080, 8085, 8051, 2270, 2280, 14IN, DSDD).

## D Line format

Lines are edited in field format as follows:

The last data statement in the BASIC program has, among other things, the tab positions for the edit line. There are four fields and with the tab stops set at 1,10,20,51,123 they have the following meanings:

Field	Usual meaning	Length
1	tag field	8
2	opcode	9
3	operand	30
4	comment	71

The length of each field is constant and changeable only with a change of tab stops.

An asterisk at the beginning of the line, defines the whole line to be a comment whose maximum length is 123. Field orientated features, except for tab and reverse tab, are inactive for a comment line.

If the first character of a line is not an asterisk, the fields are as defined in the chart.

The line currently being edited is preceded by a right pointing triangle and the field being edited is underlined. The cursor is displayed at the character entry point. There is always at least one blank separating the fields.

## E Initiating the Editor

Since the editor uses a work file on a disk, some conventions must be observed to prevent multiple users of the editor and the same work disk from using the same work file. This is usually solved by using the user's initials (two characters) and his (her) partition number.

When logging on, the users initials are asked for. Then all other users on the same work disk with the same initials are displayed and may be logged off. The users log-on tag is displayed and may be edited. From the position in the log-on file the user is assigned a unique workfile.

When the prompt is CODE; leave the field blank if no syntax checking is to be done (or if the syntax for the language being edited is not supported). As of this date the following syntaxes are supported: Z80, 8080, 8051, 8085, 2270, 2280, 14IN, DSDD. If CODE was not answered with a blank, a prompt LOGICAL EXPRESSION is shown. See the appropriate assembler memo for explanation for this.

The name of file to be edited is asked for, what disk it should be loaded from, and whether or not to load the file.

## F Functions other than Edit

- |          |                    |   |
|----------|--------------------|---|
| SF'00    | <u>MENU/EDIT</u>   | Toggles between a menu listing the functions and edit   |
| SF'01    | <u>DISK</u>        | Most disk operations (SAVE, insert LOAD, and CREATE file) are done with this function.  |
| SF'02    | <u>MOVE/COPY</u>   | Moves (physically moves) or copies (replicating) the specified line(s) to an other place in the file and automatically inserts the proper amount of space.  |
| SF'03    | <u>CAP. LOCK</u>   | Allows the A/A, A/a switch to be in the lower case and have the appropriate fields still be in upper case. Four fields are represented, a 'U' means upper case only and an 'L' means upper or lower case allowed. |
| EDIT key | <u>LINE REQ.</u>   | Allows quick access to a specific line.   |
| SF'16    | <u>LOGOFF</u>      | Logs the user off the editor and releases the work file he (she) was using.   |
| SF'17    | <u>CHANGE NAME</u> | The name of the file currently in the work area or the name of the file to be loaded may be changed along with the disk address and the work area may be cleared and loaded with the file named earlier.          |

SF'18     SEARCH       Searches for the specified input from the current line and stops when the input is found. By keying the RUN key the searched-for input is replaced by the specified replacement. By keying the return key the next occurrence of the input is searched for.

SF'19     PRINTER LIST   Lists the selected lines to the selected printer.

G   Editing the file

The following keys have special meaning while editing:

TAB (FN)	Non-destructively moves the cursor to the next field
TAB (shifted)	Reverse tab; non-destructively moves the cursor back one field
RETURN	Enters the line currently being edited (the line the cursor is on) into the work area and checks the syntax (if this option is enabled)
RUN	Does the same thing as RETURN but also inserts a blank line under the line currently being edited (this line is not really there until a RETURN is done on it)
space bar	Destructively moves the cursor to the next field, except in a comment field
CONTINUE	Changes the value of the next keystroke to an alternate value (space to a non-tabbing space, up arrow to a left arrow, slash to backslash) and as the second character in a blank comment line, the next keystroke is flooded into the line
BACKSPACE	Destructively moves the cursor one character space to the left (jumps across field boundaries)

The following applies to all keyboards except 2236DW

LINE ERASE	erases the whole line but does not delete it
CLEAR	Deletes the line currently being edited and moves the higher numbered lines up to fill the gap
recall	recall the line currently being edited to what it was prior to any keystroke unless the keystroke moved the cursor off the line
insert (lc)	puts the editor into insert mode or takes it out of insert mode
INSERT (uc)	allows insertion of a number of lines
delete (lc)	deletes a single character
DELETE (uc)	allows deletion of a number of lines
erase (sf'8)	erases the line starting from the cursor
ERASE (sf'24)	deletes the line currently being edited and moves the lower numbered lines down to fill the gap (similar to CLEAR)
begin (lc)	puts the cursor at the beginning of the line currently being edited
BEGIN (uc)	puts the cursor at the beginning of the file currently in the work area
end (lc)	moves the cursor to the end of the line currently being edited
END (uc)	moves the cursor to the end of the file currently in the work area
up arrow (lc)	moves the cursor up one line
UP ARROW (uc)	moves the cursor up one page
dn arrow (lc)	moves the cursor down one line
DN ARROW (uc)	moves the cursor down one page
left 5	moves the cursor left 5
left 1	moves the cursor left 1
right 1	moves the cursor right 1
right 5	moves the cursor right 5

The following applies to the 2236DW keyboard

ERASE(uc)	erases the whole line but does not delete it
erase(lc)	Deletes the line currently being edited and moves the higher numbered lines up to fill the gap
recall	recall the line currently being edited to what it was prior to any keystroke unless the keystroke moved the cursor off the line
insert (lc)	puts the editor into insert mode or takes it out of insert mode
INSERT (uc)	allows insertion of a number of lines
delete (lc)	deletes a single character
DELETE (uc)	allows deletion of a number of lines
PREV SCRNL(uc)	puts the cursor at the beginning of the file currently in the work area
prev scrn(lc)	moves the cursor up one page
NEXT SCRNL(uc)	moves the cursor to the end of the file currently in the work area
next scrn(lc)	moves the cursor down one page
WEST ARROW(uc)	puts the cursor at the beginning of the line currently being edited
west arrow(lc)	moves the cursor left 1
EAST ARROW(uc)	moves the cursor to the end of the line currently being edited
east arrow(lc)	moves the cursor right 1
up arrow	moves the cursor up one line
dn arrow	moves the cursor down one line

