

SECTION 4  
AIM 65 TEXT EDITOR

The AIM 65 Text Editor allows you to create and edit files of text. These files are most commonly used as the source program into the R6500 Assembler, and are written in R6500 assembly language (see Section 5). The Text Editor files are also used to store messages that are operated on by user written programs. Another use is that of general documentation.

The Editor is used by entering data into a Text Buffer from an input device, editing the data, then either storing the data on an output device or operating directly on the data using another AIM 65 or user program. Usually the AIM 65 keyboard is used to input data and an audio cassette recorder is used for storing data. Editing is performed from the keyboard. The most common output devices are the display/printer or printer for hardcopy printout and the audio cassette for permanent data storage.

It is possible that a given program may be too long to store and edit in the Text Buffer at one time. In this case, the total program should be divided into several smaller files. The Assembler has a .FILE directive that allows individual source files to be linked together to form a total source program. See Section 5.8.8 for details.

The process of dividing large programs or data into smaller segments (modules) provides a convenient method for data handling and editing.

UNIVERSITY

#### 4.1.1 AIM 65 Text Editor Features

The features of the AIM 65 Text Editor include:

Complete User Control of Memory -- The user may establish the text buffer anywhere in available RAM.

Input Device Flexibility -- The input to the Editor may come from any of several peripheral devices:

- AIM 65 or TTY keyboard
- Audio cassette
- TTY punched paper tape
- User defined device

Output Device Flexibility -- The output from the Editor may be directed to any of several peripheral devices:

- AIM 65 display/printer
- AIM 65 printer only
- Audio cassette
- TTY punched paper tape
- User defined device

Flexible Editing Commands -- Simple single character commands provide quick and easy edit functions:

Go to top or bottom of text

Go up or down a line

Find a character string

Change a character string to a different character string

List one or more lines to an output device

Insert one line from the keyboard

Read one or more lines from an input device

Delete a line

Display the current line

Quit the Editor

Re-enter and Re-edit Capability -- The Editor also provides a re-entry capability so that previously entered text may be re-edited to correct errors. This is especially useful when assembling: a source program may be entered in the Text Editor, assembled to identify any coding errors and the Editor re-entered to correct the errors. When an error-free assembly is attained, the source file may be permanently saved on audio cassette.

#### 4.1.2 Text Buffer

The text is stored in a user-specified area of RAM called the Text Buffer. Upon initial entry into the Editor, the user must define the starting and ending limits of the Text Buffer. If it is desired to allocate all of available memory to the Text Buffer, default limits determined by AIM 65 may be used. The default lower limit is \$0200. This bypasses RAM page 0, which is reserved for user and AIM 65 data, and page 1, which is reserved for the user and AIM 65 stack and for AIM 65 data. The default upper limit is

REVISIONS

determined by the amount of contiguous installed RAM, starting with address \$0200. AIM 65 checks for the existence of a page of RAM by performing a write and read test every \$100 addresses.

Data is stored in the Text Buffer in ASCII format (see Appendix E for the ASCII code format). Each character entered requires one byte (8 bits) of RAM. The text is also stored in variable length lines. Each text line ends with a RETURN (ASCII \$0D) which is stored after the last text character. Line Feed (ASCII \$0A) characters are not stored.

A Null character (ASCII \$00) is stored after the last text line to indicate the end of active text. Care should be taken that the character \$00 is not inadvertently stored in the active text area of the Text Buffer. If a \$00 is stored in the active text area, all the text past the \$00 will not be available to the user using normal editor commands (see Editor Data Recovery Techniques, Section 4.1.3).

To estimate the amount of RAM required for the Text Buffer: allow one byte for each text character, one byte for each line terminated with a RETURN (ASCII \$0D) and one byte for the text end character (ASCII \$00). Additional memory should be allocated to allow for text additions and changes.

The actual starting and ending addresses of the active text in the Text Buffer as well as the Text Buffer ending address can be determined by examining the following dedicated memory locations:

<u>ADDRESS</u>	<u>PARAMETER</u>	<u>EXAMPLE</u>
00E1	Text Ending Address Low	\$42
00E2	Text Ending Address High	\$03
00E3	Text Starting Address Low	\$00
00E4	Text Starting Address High	\$02
00E5	Text Buffer Ending Address Low	\$00
00E6	Text Buffer Ending Address High	\$04

Figure 4-1 illustrates this example.

#### 4.1.3 Editor Data Recovery Techniques

Data in the Text Buffer may appear to be lost due to two types of inadvertent actions:

- Initialization of the Editor using the Monitor E command before data in the old Text Buffer is permanently stored.
- Storing of the text end character (\$00) somewhere in the active text area.

#### INADVERTENT INITIALIZATION RECOVERY

When the Text Buffer is initialized with the Monitor E Enter Editor command, only one byte of text in the old text area may be changed. This will occur only if the new Text Buffer starting address is the same as the old starting address or is somewhere in the old active text area. The

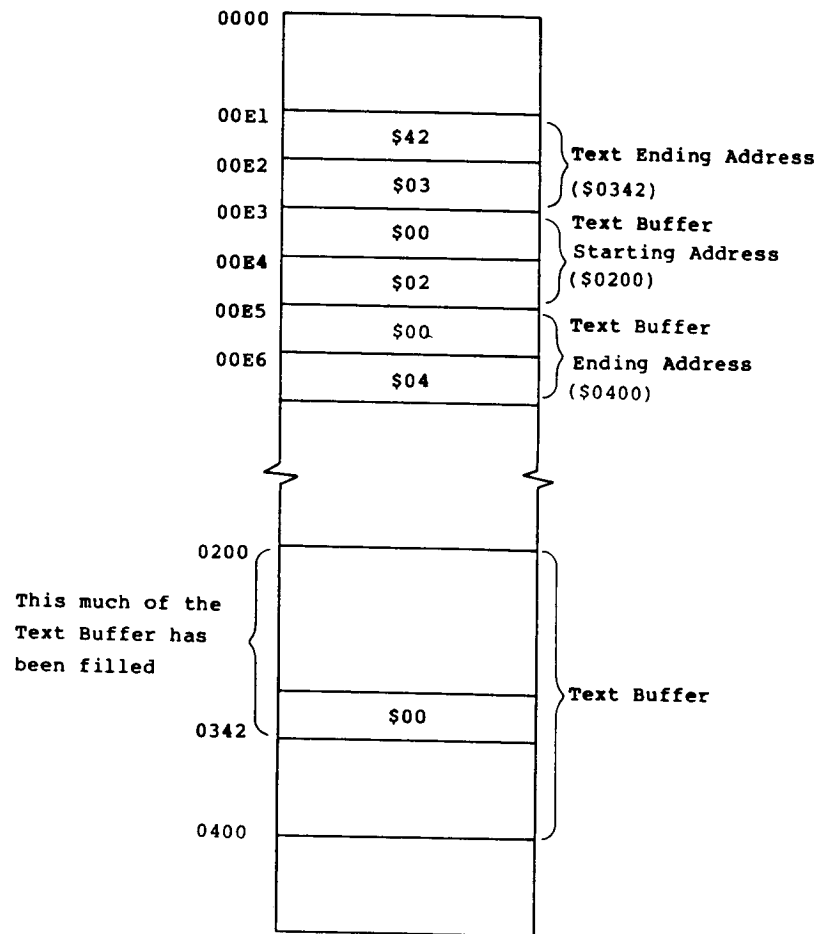


Figure 4-1. Example Text Buffer in RAM

remaining old text will still be intact, but a little difficult to find. This assumes, of course, that AIM 65 power has not been turned off or other data has been loaded into the old active text area in the meantime.

In order to recover the old Text Buffer contents, the old active text conditions must be reconstructed.

The recovery procedure is:

1. Initialize the Editor with the Monitor E command and set the Text Buffer starting "FROM=" and ending "TO=" address limits to the same value as the old buffer.
2. Using the Monitor M and / commands, change the \$00 value located at the address specified by the starting address (\$00E3 and \$00E4) to a temporary valid ASCII character, e.g., \$41 for "A". After the Text Buffer is completely recovered, the "A" can be easily changed back to its correct value.
3. When the Editor was inadvertently initialized, the Editor stored a \$00 at the address specified by the new Text Buffer starting address to indicate an unfilled buffer. If this \$00 is in the old active text area and is at an address different from the \$00 in Step 2, it too must be located and changed to a temporary valid ASCII text character code. This character can also be easily changed to its proper value later. Use the Monitor M and SPACE commands to scan memory starting with the address specified by

the current Text Buffer starting address (\$00E3 and \$00E4). If data has been read into the current Text Buffer, the \$00 may be located further down in the Text Buffer.

4. Using the Monitor M and SPACE commands, scan memory starting with the Text Buffer starting address specified in \$00E3 and \$00E4, locate the old end of active text indicator (\$00). Enter the located address into the Text Buffer ending address (\$00E1 and \$00E2).
5. Re-enter the Editor using the Monitor T command. The top line of the data in the old Text Buffer should be displayed. Go to bottom of the data using the Editor B command. The last line of data in the old Text Buffer should be displayed. If not, one of the recovery steps was not completed properly.
6. Find any temporary ASCII characters loaded into memory in place of \$00 to help recover the old Text Buffer. Change them to the desired values.

Note that the \$00 character will be found immediately after a \$0D end of line indicator. If the \$00 is not located, the \$00 can be entered into memory after any \$0D character and that address entered as the Text Buffer ending address (\$00E1 and \$00E2) to help locate the old end of text.

#### TEXT BUFFER RECOVERY PROGRAM

A program may be entered using the I command that will automatically perform the Text Buffer recovery. This program performs the steps discussed in the previous paragraph. The program starts at \$03D0 in order to place it at a high RAM value in the 1K RAM version. By changing the high address value from "03" to "0F" in the program counter and conditional branch operand fields, the program can be moved to high RAM in the 4K version.

The program can easily be typed in when needed or may be typed in and then recorded on audio cassette in object code form using the Monitor D command for future use, if required. The program operates by changing the text ending address to the address of the first \$00 located past the initial \$00 at the text starting address. As such, the program may be run more than once if the first run found another \$00 which was not the expected end of text indicator.

Type in the Text Buffer recovery program as follows:

```
<I>
F474      +=03D0
03D0 AD   LDA 00E3
03D3 8D   STA 00E1
03D6 AD   LDA 00E4
03D9 8D   STA 00E2
03DC A9   LDA #00
03DE A8   TAY
03DF A9   LDA #41
03E1 91   STA (E1),Y
03E3 98   TPA
03E4 D1   CMP (E1),Y
03E6 F0   BEQ 03F1
03E9 E6   INC E1
03EA D0   BNE 03E4
03EC E6   INC E2
03EE 4C   JMP 03E4
03F1 4C   JMP E1R1
```

COMPUTER

Verify the program by disassembling 16 instructions using the K command.

### RUNNING THE TEXT BUFFER RECOVERY PROGRAM

1. Initialize the Text Editor to the initial Text Buffer limit starting and ending addresses, then return to the Monitor using ESC.
2. Enter the Text Buffer recovery program using the Monitor I command or load from audio cassette using the Monitor command.
3. Set the program counter to the starting address of the Text Buffer Recovery program.

For example:

\*=\$03D0

4. Type G/. to execute in the program. When the program finds \$00, it will display the Monitor prompt.

#### NOTE

If a \$00 is not found, the program may hang up. Return to the Monitor by depressing RESET.

5. Type T to re-enter the Text Editor at the top of the Text Buffer. Change the first character as appropriate.

### EXAMPLE OF TEXT BUFFER RECOVERY USING THE TEXT BUFFER RECOVERY PROGRAM

CE)		CT)
EDITOR		END
FROM=0200 TO=0300		=CG)
IN=		
LINE 1		<*)=03D0
LINE 2		<G)/.
LINE 3		CT)
=CG)		AINE 1
=CG)		=CL)
CT)		/.
LINE 1		OUT=
=CL)		AINE 1
/.		LINE 2
OUT=		LINE 3
LINE 1		
LINE 2		=CT)
LINE 3		AINE 1
		=CG)
END		A
CE)		AINE 1
EDITOR		TO=L
FROM=0200 TO=0300		LINE 1
		=CT)
IN=		LINE 1
END		=CL)
=CG)		/.
		OUT=
		LINE 1
		LINE 2
		LINE 3

LOW DUTER

#### 4.1.4 Line Pointer

The AIM 65 Text Editor is a line oriented editor; all Editor operations begin from the start of the active line. The active line is identified by the line pointer, which is always positioned in front of the first character of the active line. After an Editor operation is performed, the line pointer is positioned to the start of either the last line operated on or one line down from the last line operated on, depending upon the command.

The active line is displayed at the completion of most Editor commands, depending on output device selection. If there is any doubt where the line pointer is positioned, the Editor SPACE command will display the active line.

Line pointer positioning commands allow easy manipulation of the line pointer. Using these commands, the line pointer can be easily and quickly moved to the top of the text, to the bottom of the text, up one line, or down one line.

#### 4.1.5 Dummy Line

A dummy line is provided after the last active text line to allow text to be added at the end of the buffer. If the line pointer is positioned on the last active line of text in the Text Buffer, it must be moved down one line to the dummy line using the D command to read or insert new text at the end of the active text.

When the line pointer is positioned on the dummy line, either "END" or no data will be displayed.

An L command listing all the active text will leave the line pointer positioned on the dummy line.

#### 4.2 EDITOR ENTRY AND EXIT COMMANDS

Two commands permit the Editor to be entered from the Monitor. One command initializes the Text Buffer upon entry, the other allows re-entry to the Editor without affecting the text stored in the Text Buffer. Two separate commands allow return to the Monitor: one is used to quit the Editor during an Editor idle state, the other is used the escape during Editor command execution.

##### 4.2.1 E Command - Enter and Initialize the Editor

The Monitor E command is used to enter the Editor, initialize the Text Buffer, and to read data into the text buffer.

Use the E command as follows:

1. After the Monitor prompt, type E. AIM 65 will enter the Editor mode and respond with:

```
E
EDITOR
FROM=
```

Table 4-1. AIM 65 Text Editor Commands

<u>CATEGORY</u>	<u>COMMAND</u>	<u>FUNCTION</u>
Editor Entry and Exit	E	Enter and Initialize Editor
	T	Re-Enter Editor
	Q	Quit Editor and Re-Enter Monitor
	RESET	Enter and Initialize Monitor
	ESC	Re-Enter Monitor
Text Input/ Output and Update	R	Read Into Text Buffer
	L	List From Text Buffer
	I	Insert One Line
	K	Delete One Line
Line Pointer Positioning and Display	T	Move the Line Pointer to the Top
	B	Move the Line Pointer to the Bottom
	U	Move the Line Pointer Up One
	D	Move the Line Pointer Down One
	SPACE	Display Current Line
Character String	F	Find Character String
	C	Change Character String

- Enter the text buffer starting address as a hexadecimal number, followed by a RETURN or SPACE. If more than four digits are entered (up to 11) AIM 65 will use only the last four. Entering RETURN or SPACE without a start address will cause the default value of \$0200 to be used. If the default address is used or \$0200 is entered, AIM 65 will respond with:

FROM=0200 TO=

- Enter the Text Buffer ending address as a hexadecimal number, followed by a RETURN or SPACE. If more than four digits are entered (up to 11), AIM 65 will use only the last four. Entering RETURN or SPACE without entering an ending address will cause the last address of contiguous installed RAM (starting with address \$0200) to be used as the ending address.

The Text Buffer is allocated from the start address specified through the ending address specified. A write/read memory check is made at an address of each page of memory to make sure the specified memory space is available. If an ending address was entered and the memory write fails, the message MEM FAIL will be printed, indicating the first address of the memory page that failed to write, and the system will return to the Monitor. The memory write test is performed every \$100 addresses, starting with the entered FROM address. For example, if FROM=0250 and TO=0440, the test is performed at \$0350 and \$0450. If memory is



installed from \$0000 to \$03FF, the memory test will fail at location \$0450 with the message:

<MEM FAIL 0400

If an ending address was not specified, i.e., the default ending address used, and the memory write fails, the ending address of the Text Buffer is established as the ending address of the failed page. For example, in the 1K RAM version of AIM 65 the memory write will fail at address \$04FF. The default ending address will, therefore, be set at address \$0400.

If the default address in the 1K version of AIM 65 is used, or address 0400 is entered, AIM 65 will respond with:

FROM=0200 TO=0400  
IN=

4. After the Text Buffer is initialized, the E command allows the user to go directly into the Editor read mode by asking for the input device. Enter the code of the input device from which the text is to be entered.

<E>  
EDITOR  
FROM=0200 TO=0400

IN=  
TOP LINE IN BUFFER  
LINE 2  
LINE 3  
BOTTOM LINE

Any time a serious error occurs while using the Editor, the user may return to the Monitor Mode by pressing ESC or depressing the RESET button. The Editor can be re-entered using the T command without any loss of input data.

#### 4.2.2 T Command - Re-enter the Editor

The T command is used to re-enter the Editor from the Monitor, without altering the text in the Text Buffer. The Line Pointer is automatically positioned at the top line.

Use the T command by typing T as response to the Monitor prompt. AIM 65 will respond with:

<T>  
THE TOP LINE IS DISPLAYED

#### NOTE

The T command will not operate properly unless the Text Buffer has been previously initialized with the E command.

Example:

=<T>  
TOP LINE OF TEXT

#### 4.2.3 Q Command - Exit the Editor and Re-Enter the Monitor

The Q command is used to exit the Editor and return to the Monitor.

Use the Q command by typing Q. AIM 65 will respond with:

=<Q>  
<

The Monitor is now active as indicated by the Monitor prompt and monitor commands may be entered.

#### 4.2.4 ESC Command - Re-enter the Monitor

The ESC command is used to escape from the Editor and return to the Monitor. The ESC key is examined by the Editor along with the other command keys to determine if a valid Editor command has been entered. A check to determine if the ESC command has been entered is also made at the end of each line of text during the L command (see Section 4.3.4).

Escape from the Editor by typing ESC. AIM 65 will respond with the Monitor prompt:

<

The Monitor is now active and Monitor commands may be entered.

#### NOTE

If an L command (List from Text Buffer) is in progress, the check for entry of the ESC command is made only at the end of each line that is listed. The ESC key must be held down long enough for the Editor to sample it in this case.

#### 4.2.5 RESET Command - Enter and Initialize the Monitor

The RESET command is used to interrupt the Editor at any point of operation and re-enter the Monitor. The RESET command causes a hardware interrupt to occur, so any Editor operation in progress will be terminated without completion. See Section 3.2.6 for a complete description of the RESET command.

Use the RESET command by pressing the RESET button. AIM 65 will respond with:

ROCKWELL AIM 65

The Monitor is now active and Monitor commands may be entered.

#### 4.3 TEXT INPUT/OUTPUT AND UPDATE

Four commands allow text to be read into, listed from, and deleted from the Text Buffer on a single- or multiple-line basis.

##### 4.3.1 R Command - Read Lines Into Text Buffer

The R command is used to read multiple lines from an input device into the Text Buffer. Text read into the text buffer is inserted in front of the active line.

There may be a noticeable pause at the end of each line if text is being read in front of already existing lines.

If any attempt is made to read more text than can be stored into the Text Buffer, an END message will be displayed.

If this occurs, the Buffer must be expanded by changing the Text Buffer ending address (in \$00E5 and \$00E6) to a larger value using the Monitor M and / commands.

Use the R command as follows:

1. Type R. AIM 65 will ask for the input device by displaying:

=<R>

IN=

2. Enter the code of the input device from which the text will be entered.

- A. If the keyboard code (RETURN or SPACE) is entered, AIM 65 will display a character position cursor, (^) to indicate where the next digit will be entered:

IN=  
^

Start entering text from the keyboard, terminating each line with RETURN. An input error may be corrected by entering DEL and re-entering the desired character.

Up to 60 characters may be entered on a line. The first 20 characters are entered from left to right as seen on the display. The cursor will disappear when the 20th character is entered. Starting with character 21, the displayed data will scroll to the left one character position as each new character is entered. Upon entry of the 21st character, the first 20 entered characters will print. Likewise, upon entry of the 41st character, characters 21 through 40 will print. Characters 41 through the end of entered text will print when RETURN is entered.

The final line should be terminated with two RETURNS, which will end the Keyboard Read function and allow the Editor to accept a new command.

ATN COMPUTER

NOTE

After character 21 or 41 is entered, time must be allowed for the line to be printed before more data is entered. Since the keyboard is not scanned during printing, any character entered during this period may be lost. At normal typing speed no data should be lost; however, if data is typed in rapidly, additional time should be allowed.

- B. If the input is from AIM 65 audio tape, see Section 9.1.6. If the input is from TTY punched tape, see Section 9.2.3 for the detail procedure.

When the file read is completed, END will be displayed.

Example:

```

=END
LINE
LINE
LINE
LINE
LINE
LINE
LINE
LINE
LINE
LINE

```

4.3.2 I Command - Insert One Line

The I command is used to insert one line of text ahead of the active line. Input is always from the keyboard.

Use the I command as follows:

1. Use the T, B, U, D, or F command to locate the desired active line.
2. Type I. AIM 65 will respond with:
 

```

=<I>
^

```
3. Enter the line of text to be inserted. Terminate the entry with a RETURN. AIM 65 will display the next line down.

Example:

Suppose the program in the Text Buffer is (using the T and L commands):

```

=END
TOP LINE OF TEXT
=END
^
OUT=
TOP LINE OF TEXT
LINE 2
LINE 3
LINE 4
BOTTOM LINE

```

COMPUTER  
TTC

It is desired to add a line before the third line. Using the T, F, and I commands, the new line is inserted:

```
=<TD  
TOP LINE OF TEXT  
=<FD  
2  
LINE 3  
=<ID  
LINE 2A  
LINE 3
```

After inserting the new line of text, the updated program is (using the T and L commands):

```
=<TD  
TOP LINE OF TEXT  
=<LD  
A  
OUT=  
TOP LINE OF TEXT  
LINE 2  
LINE 2A  
LINE 3  
LINE 4  
BOTTOM LINE
```

#### 4.3.3 K Command - Delete One Line

The K command is used to delete the active line of text.

Use the K command as follows:

1. Locate the line of text to be deleted using the T, B, U, D, or F command.

2. Type K. AIM 65 will respond with:

```
=<K>  
DISPLAY OF NEXT LINE DOWN
```

Example:

Assume the program in the Text Buffer is (using the T and L commands):

```
=<TD  
TOP LINE OF TEXT  
=<LD  
A  
OUT=  
TOP LINE OF TEXT  
LINE 2  
LINE 2A  
LINE 3  
BOTTOM LINE
```

Locate the line to be deleted (line 2A) using the T and F commands, then delete it with the K command:

```
=<TD  
TOP LINE OF TEXT  
=<FD  
2A  
LINE 2A  
=<KD  
LINE 2A  
LINE 3
```

COMPUTER

After deleting the desired line of text, the updated program is (using the T and L commands):

```
=<T>
TOP LINE OF TEXT
=<L>
/
OUT=
TOP LINE OF TEXT
LINE 2
LINE 3
BOTTOM LINE
```

#### 4.3.4 L Command - List Lines From Text Buffer

The L command is used to list a specified number of lines in the Text Buffer to an output device, starting at the beginning of the active line.

#### CAUTION

If the text is to be listed to cassette and read back into a partially-filled text buffer, the interblock gap size in \$A409 must be increased to \$80.

Use the L command as follows:

1. Type L. AIM 65 will respond with:

```
=<L>
/
```

2. Specify the number of lines to be listed by entering a decimal number from 01 to 99; 00 lists 100 lines, RETURN lists one line, a . or SPACE lists all the lines in the Text Buffer. AIM 65 will ask to which device type the text is to be listed:

OUT=

3. Enter the code of the device to which the text is to be listed:

<u>DESIRED OUTPUT DEVICE</u>	<u>TYPE</u> <u>DEVICE CODE</u>	<u>ADDITIONAL</u> <u>PROCEDURE</u>
AIM 65 Display/ Printer	RETURN or SPACE	
AIM 65 Printer	P	
Audio Tape - AIM 65 Format	T	See Section 9.1.2
TTY Punched Tape	L	See Section 9.2.2
User Defined	U	See Section 7

4. AIM 65 will then proceed to list the contents of the Text Buffer, beginning with the active line and ending with the specified line, to the specified output device. If the specified output device is the audio cassette recorder, a count of each 80 character block will be indicated as it is listed.

COMPUTER

CAUTION

If the listed output is to be recorded on audio cassette tape for subsequent input to the assembler or as added text to a partially loaded Text Buffer, the TSPEED parameter in address \$A409 must be changed to \$80 to provide a large gap time between recorded data blocks. See Section 7.6.

To save the entire contents of the Text Buffer on tape, the T command should be used to position the line pointer at the beginning of the Text Buffer and the L/ or L/SPACE commands used to list all the text lines in the Text Buffer.

Example:

```
=<T>
TOP LINE OF TEXT
=KLT
/
OUT=
TOP LINE OF TEXT
LINE 2
LINE 3
BOTTOM LINE OF TEXT
```

4.4 LINE POINTER POSITIONING AND DISPLAY

Five commands allow moving the line pointer to the top or bottom of the Text Buffer, moving the line pointer up or down one line, and displaying the contents of the active line. In all cases, the line pointer is positioned at the beginning of the line after each command.

4.4.1 T Command - Move The Line Pointer To The Top

The T command is used to move the line pointer to the top line of the text in the Text Buffer and to display the line contents.

If the Text Buffer has been initialized with the E command and no text has been read into the buffer, END will remain displayed.

Use the T command as follows:

Type T. AIM 65 will position the line printer to the top line of the Text Buffer and will display the contents of that line.

Example:

```
=<T>
TOP LINE OF TEXT
```

4.4.2 B Command - Move the Line Pointer to the Bottom

The B command is used to move the line pointer down to the last line of text in the Text Buffer and to display the line contents. If no data has been entered into the Text Buffer, END will be displayed.

To append text to the end of the text currently in the Text Buffer, the line pointer must be moved down one line after the B command is used. Following the B command, the D command will cause the line pointer to be moved down one

COMPUTER

line, to a dummy line following the last line of actual text, and to display END. The R or I command may be used at this point to read or insert text ahead of the dummy line.

Use the B command as follows:

Type B. AIM 65 will respond by positioning the line pointer to the last line of text and will display the contents of that line.

Example :

```
=<B>
BOTTOM LINE
```

#### 4.4.3 U Command - Move the Line Pointer Up One Line

The U command is used to move the line pointer up one line and to display the contents of that line. If an attempt is made to ascend past the top line of text, the line pointer will remain on the top line.

Use the U command as follows:

Type U. AIM 65 will respond with:

```
=<U>
DISPLAYED TEXT LINE
```

Example:

```
=<D>
LINE 3
=<UD>
LINE 2
```

#### 4.4.4 D Command - Move the Line Pointer Down One Line

The D command is used to move the line pointer down one line and to display the contents of that line. If the line pointer is moved down one line from the last line of text, AIM 65 will display END.

Use the D command as follows:

Type D. AIM 65 will respond with:

```
=<D>
DISPLAYED TEXT LINE
```

Example:

```
=<D>
LINE 3
=<DD>
LINE 4
```

#### 4.5 CHARACTER STRING

Two commands allow text location and manipulation by entry of a character string. Using these commands text can be located, or located and changed, without knowing where the text is located in the Text Buffer.



COMPUTER

#### 4.5.1 F Command - Find Character String

The F command is used to find a specified character string, of up to 19 characters long. The search will start at the beginning of the active line and continue until the first occurrence of the string is found or the end of the Text Buffer is encountered. This command is very useful to locate a certain line of text to delete, or a text line for reference prior to an insert or read command, or to determine if a certain character string exists in the Text Buffer.

Use the F command as follows:

1. Type F. AIM 65 will respond with:  
  
=<F>
2. Enter the character string that is to be found. End the input with RETURN. AIM 65 will display and search for the entered string. If line 2 had been entered, the response will be:  
  
=<F>  
LINE 2
3. A. If the string is found, AIM 65 will display the line that contains the entered character string, and position the line pointer at the beginning of

that line. If the entered character string "Line 2" is found in a line containing "Line 2 of Text", the response will be:

```
=<F>  
LINE 2  
LINE 2 OF TEXT
```

- B. If the string is not found, an END message will be displayed.

The search can be restarted by entering the T command (moving the line pointer to the top of the Text Buffer) and re-entering the entire F command.

- C. If the string is found but is not on the desired line, the search can be continued by entering F followed by RETURN after display of the cursor without re-entering the character string. The line pointer will be positioned at the beginning of the next line and the search will continue as described above.

Example:

```
=<T>  
TOP LINE OF TEXT  
=<F>  
2A  
LINE 2A
```

#### 4.5.2 C Command - Change Character String

The C command is used to search for a given character string (up to 19 characters long), and either delete or alter it to a character string of up to 20 characters long. The search will start at the beginning of the active line, and continue until the entered string is found or the end of the Text Buffer is encountered.

Use the C command as follows:

1. Type C. AIM 65 will respond with:

=<C>

2. Enter the string that is to be changed, and end the input with RETURN. If string LAB 2 had been entered, the response will be:

=<C>

LAB2

3. The line containing the first occurrence of the string will be displayed, and the Editor will wait for an indication that the displayed line contains the desired occurrence of the string. If the string LAB 2 is found in a line containing LAB2; COMMENT, the response will be:

LAB2 ;COMMENT

If the entered string is not found, the AIM 65 will display an END message.

The change can be restarted by entering the T command (to move the line pointer to the top of the Text Buffer) and re-entering the entire C command.

4. If the displayed line does not contain the correct occurrence of the string, press any key except RETURN to resume the search.
5. When the correct occurrence of the string is found, press RETURN to terminate the search. AIM 65 will respond with:

TO=^

6. Enter the new character string, ending the input with a RETURN. The old text string will be replaced with the new text string.

If the old string is to be deleted rather than replaced, enter RETURN immediately after the cursor is displayed. If LAB2 is to be changed to NOP, enter NOP. AIM 65 will respond with:

TO=NOP

Once RETURN is entered, the updated line of text will be displayed:

NOP ;COMMENT

COMPUTER

Example:

```
=<TD  
TOP LINE OF TEXT  
=<CD  
INK  
LINK 4  
TO=INE  
LINE 4  
=<TD  
TOP LINE OF TEXT  
=<LD  
/  
OUT=  
TOP LINE OF TEXT  
LINE 2  
LINE 3  
LINE 4  
BOTTOM LINE
```