



## Dissecting C. W. Moser's ASSM/TED 1.0

Francis Turco

Carl Moser's excellent assembler/text editor for the 6502 Microprocessor has been reviewed superficially in several publications.<sup>1,2</sup> So far, no one has done an in-depth write-up for a PET owner who wants to understand or modify his copy. The manual provided by Moser is adequate, but sketchy in some areas. I, for one, would like to see some articles by users who have figured out solutions to problem areas.

For example, PET owners find out (on page 35 of the manual) that "At present, the ASSM/TED does not contain a printer subroutine...". In another area, the ASSM/TED is designed for a "standard" PET and utilizes the audio cassette drives for off-line storage. The manual (Sections 6 & 7) discusses configuring the ASSM/TED for disk operation and using it with disk. This discussion is too brief to be understandable by a novice assembly language programmer.

In still another area, the editor has many powerful capabilities and will accept a full line of characters (65 typed characters) but the sense of the shift key is reversed. That is, shift gives lower case letters. Unshifted gives upper case letters. This proves to be cumbersome when typing a letter or manuscript from the PET keyboard.

In an effort to shed some light for others, who like myself, are trying to understand and modify their copy of ASSM/TED and perhaps stimulate some of you to share your findings, I am submitting some areas that I have uncovered in Moser's Assembler.

Figure 1 shows a memory map of the assembler/text editor. The assembler is written for a 16K PET and fills almost all useable memory space. As the figure shows, the assembler and text editor are co-resident and occupy the space from \$2000 thru \$3FFF. Commodore's monitor occupies the area from \$0400 thru \$076C. This leaves enough memory for a relocatable file (\$1F00 thru \$1FFF), a lable file (\$1800 thru \$1EFF), and approximately 4K for user programs (\$0770 thru \$17FF).

Table I is a list of addresses of major routines. This is a fun table — try some experimenting with it. For example, RUN 8390 will assemble your program. RUN 8390 LIST will assemble and list. RUN 8470

will print your program. Table II provides a list of addresses of the pseudo opcode routines, while Table III contains some interesting areas that will be helpful to someone modifying his assembler.

Carl Moser's ASSM/TED is a very good program and will allow the PET owner to convert his PET into a 6502 development station with a little effort on his part. If the PET is equipped with a line printer off the IEEE port, the owner can easily get around the first problem area and get a listing of his source code and/or his assembly. This subject will be treated in PART II of this article.

1. **Compute**, Fall 1979, p. 100, "6502 Macro Assembler and Text Editor SYM Version" by Harvey Herman
2. **The PAPER**, Vol. II, Issue 6, August 1979, "Relocating Macro Assembler/Text Editor 1.0 by R. Busdieker

*Figure 1. ASSM/TED 1.0 Memory Map*

HEX	DEC	
3FFF 2000	16383 8192	ASSEMBLER & TEXT EDITOR by C. W. MOSER
1FFF 1F00	8191 7936	RELOCATABLE FILE (256 BYTE BUFFER)
1EFF 1800	7935 6144	LABEL FILE (SYMBOL TABLE)
17FF 0770	6143 1904	USER'S TEXT FILE (SOURCE CODE)
076C 0400	1900 1024	COMMODORE'S MONITOR (876 BYTES)
03FF 0000	1023 0	RESERVED FOR COMMODORE'S OPERATING SYSTEM

**Table I**  
**MAJOR ASSEMBLER**  
**ROUTINES**

HEX	DEC	ROUTINE	
2033	8243	CLEAR	user's text file

<b>208A</b>	<b>8330 BREAK</b>	to monitor
<b>2098</b>	<b>8344 AUTO</b>	line number
<b>20A0</b>	<b>8352 GET</b>	program from tape
<b>20A6</b>	<b>8358 FORMAT</b>	text file
<b>20B6</b>	<b>8374 MANUSCRIPT</b>	line numbers output/not output
<b>20C6</b>	<b>8390 ASSEMBLE</b>	source code
<b>20FF</b>	<b>8447 RUN</b>	program previously assembled
<b>2116</b>	<b>8470 PRINT</b>	text File
<b>2AFB</b>	<b>11003 OUTPUT</b>	create a relocatable object file
<b>2E52</b>	<b>11858 LABELS</b>	prints out label file
<b>31EE</b>	<b>12782 PASS</b>	execute the second pass of assembly
<b>333E</b>	<b>13118 NUMBER</b>	re-number text file
<b>3467</b>	<b>13415 PUT</b>	program out to tape
<b>3559</b>	<b>13657 FIND</b>	character string specified
<b>355F</b>	<b>13663 EDIT</b>	change source code
<b>3844</b>	<b>14404 HARD</b>	print routine (not functional on PET)
<b>3873</b>	<b>14451 COPY</b>	lines of text
<b>39B9</b>	<b>14777 MOVE</b>	lines of text
<b>39C2</b>	<b>14786 DELETE</b>	lines of text
<b>39EF</b>	<b>14831 SET</b>	boundaries of text file, label file & buffer
<b>3A80</b>	<b>14976 DUPLICATE</b>	files from tape 1 to tape 0
<b>3AB6</b>	<b>15030 ENTER</b>	file name in the diskette directory
<b>3AC7</b>	<b>15047 LOOK UP</b>	file name in the diskette directory
<b>3B50</b>	<b>15184 SHIFT</b>	upper/lower case

**Table II****PSEUDO OPCODE ROUTINES**

<b>HEX</b>	<b>DEC</b>	<b>ROUTINE</b>
<b>2919</b>	<b>10521</b>	<b>.DS</b> Designate Storage
<b>2964</b>	<b>10596</b>	<b>.EJ</b> Eject
<b>297B</b>	<b>10619</b>	<b>.RS</b> Resolve address & Store
<b>2980</b>	<b>10624</b>	<b>.CE</b> Continue with Errors

<b>2985</b>	<b>10629 .OS</b>	Object Store option
<b>298A</b>	<b>10634 .OC</b>	Object store option Clear
<b>298F</b>	<b>10639 .CT</b>	Continues on Tape
<b>2994</b>	<b>10644 .LS</b>	List option Set
<b>2999</b>	<b>10649 .LC</b>	List Option Clear
<b>299F</b>	<b>10655 .SI</b>	Store Internal address
<b>29A8</b>	<b>10664 .SE</b>	Store External address
<b>29B3</b>	<b>10675 .BA</b>	Beginning Address
<b>29F3</b>	<b>10739 .MC</b>	Move Code
<b>2A1D</b>	<b>10781 .BY</b>	Bytes
<b>2A57</b>	<b>10839 .DI</b>	Designate Internal
<b>2A60</b>	<b>10848 .DE</b>	Designate External
<b>2AB7</b>	<b>10935 .EN</b>	End
<b>3378</b>	<b>13176 .RC</b>	Resolve Code
<b>3D1E</b>	<b>15646 .ES</b>	Output macro generated object code
<b>3D23</b>	<b>15651 .EC</b>	Supress macro generated object code
<b>3D6A</b>	<b>15722 .MD</b>	Macro Definition
<b>3E0C</b>	<b>15884 .ME</b>	Macro End

**Table III****INTERESTING  
AREAS**

<b>HEX</b>	<b>DEC</b>	<b>ROUTINE</b>
<b>2000</b>	<b>8192</b>	Cold start of ASSM/TED 1.0
<b>203F</b>	<b>8255</b>	Command Line Interpreter
<b>207A</b>	<b>8314</b>	Initializes Pointer for Text File
<b>2090</b>	<b>8336</b>	Warm start of ASSM/TED 1.0
<b>2190</b>	<b>8592</b>	Same as 8599 + carriage return
<b>2197</b>	<b>8599</b>	Prints out the double slash after listing
<b>2602</b>	<b>9730</b>	Reads remainder of entered command - For Example: PRINT 100 200 or FORMAT CLEAR
<b>26AB</b>	<b>9899</b>	Jump Table for Major Assembler Routines (Commands)
<b>271C</b>	<b>10012</b>	Pseudo Opcode Table
<b>27AA</b>	<b>10154</b>	Mnemonics Table

<b>2E89</b>	<b>11913</b>	Xfers Pointer for Lable File to Zero Page Initialize Pointer for Lable File
<b>2F96</b>	<b>12182</b>	Stores a Zero Pointer + 2
<b>32DB</b>	<b>13019</b>	Prints character that is in accumulator (same function as 65490 in BASIC ROM)
<b>330B</b>	<b>13067</b>	Prints carriage return
<b>331A</b>	<b>13082</b>	Prints 2 spaces
<b>331D</b>	<b>13085</b>	Prints 1 space
<b>3323</b>	<b>13091</b>	Converts accumulator to Hex & prints it
<b>354F-3558</b>	<b>13647- 13656</b>	Permanent Copy of Value of Boundaries for Text, Label & Buffer (See also 14889)
<b>37E2</b>	<b>14306</b>	Moser suggests this location for a JSR to a line printer routine written by the user. The routine at 13019 would call this subroutine.
<b>3A29</b>	<b>14889</b>	Prints out the boundaries & the present end of data (See also 13647)
<b>3F00-3FFF</b>	<b>16128- 16383</b>	Relocated Page 1 variables
<b>3F35-3F85</b>	<b>16181- 16261</b>	Keyboard Buffer

- [Back to previous page](#)
- [See this article as it appeared in the magazine](#)
- [View this issue's table of contents](#)