

TSC TEXT PROCESSING SYSTEM

1 EG 308
run

COPYRIGHT 1977 © BY
TECHNICAL SYSTEMS CONSULTANTS, INC.
PO Box 2574
WEST LAFAYETTE, INDIANA 47906

ALL RIGHTS RESERVED

Contents

PREFACE	v
I. INTRODUCTION TO TEXT PROCESSING	1
II. COMMAND SUMMARY	13
III. REFERENCE MANUAL	17
IV. USING THE TEXT PROCESSOR	29
V. MACRO LIBRARY	31
VI. SYSTEM ADAPTIONS	39
VII. SOURCE LISTING	

Preface

The TSC Text Processing System is the most complex program released by TSC to date. With this in mind, the following recommendations should be noted by the user.

Do not expect to master the system with one reading of the manual. The entire document should be read lightly the first time through, followed by a more rigorous reading. The "Reference Manual" section is very concise and contains detailed descriptions of all of the commands of the processor. This is the section which should be studied extensively.

Since the system is so complex, many results may occur which are contrary to the user's intentions. If strange output is encountered, reread the sections of the manual covering the commands being used. As more experience is gained, the system will become an invaluable tool, but as with any complex system, it takes time to learn its operation.

I. INTRODUCTION TO TEXT PROCESSING.

This world is producing millions of words of text each day. There are words in newspapers, magazines, books, catalogs, pamphlets, letters, documents, and manuals, and they all need to be organized before publication. It would certainly be a never ending task if all text formatting and organization were to be done manually. It simply would not get done. Thanks to computers and programs called text processors, text formatting (sometimes called word processing) becomes a fairly trivial task. The text processor allows for convenient and precise page formatting and organization. The final copy becomes extremely readable and neat, which are desirable features of any printed matter.

Just what can be done with text processors? The simplest functions perform exact page fitting. In other words, if the text page should have one inch margins with a page number centered at the bottom of each page, and perhaps a special title at the top of each, the processor will automatically provide these, given the appropriate commands. Line justification is another feature provided. Several types are available which include left-hand justification (left edge straight, right edge ragged), right-hand only justification (left ragged, right straight), left and right (both edges are straight), and center justification (both edges ragged but lines centered). An extensive text processor will provide features which will allow special operations such as footnote processing. The TSC Text Processing System supports all of the above features.

To gain some insight into the use of a text processor, several specific examples will be given using the TSC Text Processor's command set. The commands used by text processors vary from system to system but many are used in the same fashion. The TSC Text Processor uses an intermixed command and text method. To issue a particular command to the processor, it is necessary to start the command in column number one of a new line and begin the command with the control character, a period ('.'). This is the method used by most of the large scale system formatters including NROFF*, which is the system the TSC Text Processor has been modeled after.

Before any specific examples are shown, a description of the 'environment' will be given. The environment refers to the basic page and formatting features which will be in effect unless otherwise specified. The initial or default environment is very important. The TSC processor, without any command information, will perform left and right justification with a line length of 65 characters (the standard 8 1/2" page line width). Page length is initially defined to be 66 lines which is the standard for 11" paper and 6 lines/inch spacing. Other initial environment features provide for the passing of blank lines to output, and

*NROFF is a text formatting program written at Bell Laboratories. It runs on many large operating systems, including the UNIX Time Sharing System.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

for any line starting with a space or spaces to create a new line with the spaces now treated as unpaddable space characters*. With the environment initialized as above, it is possible to take any text file not having special command information embedded in it and still receive a sensibly formatted output. This is an important feature which is often overlooked by many processor designers. The environment may, of course, be changed or modified at any time by the use of special commands to allow for more personalized and detailed formatting jobs.

Let's take a look at some specific commands of the TSC processor. One of the simplest commands is the center lines command, .CE N, where N is the number of lines to be centered within the current line width. To use this command, as with any of the commands, it is only necessary to place the command right before the lines it is to affect. For example:

```
: .CE 2
      The Design of Text Processors
      An Introduction
```

will cause the two lines listed to be neatly centered on the page. It can be seen that this is much easier than trying to manually calculate the correct spacing.

The initial environment was previously described. All of the parameters may be easily changed by the commands which directly affect them. One of the commands is .LN N and is used to set the current line length. To set the line length to 50, all that is necessary is a command line which reads as follows.

```
.LN 50
```

The length is now 50 columns. Another parameter easily set is the page length using the command .PL N, where N is the number of lines per page desired. Some other commands which change environment parameters include .FI and .NF which turn fill mode on and off (no fill) respectively. Fill means that as many words which will fit within the current line length are placed on each output line. This gives a straight left text edge and a slightly ragged right one. No fill simply copies the input lines directly to the output. It should be noted that 'fill' must be on for any justification to occur. The justification feature may be turned off using .NJ for 'no justification' or the type of justification may be set using .JU X. The X is the selection character and may be null which turns justification on in the mode it was previously defined; it may be R for right hand, C for centered, or N for normal (left + right). Left justification is obtained by turning 'fill' on and justification off.

*Unpaddable spaces are characters which appear as spaces on the output but are not recognized as such by the processor. This means these spaces will not be 'spread out' by the justification routines.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

Another environmental parameter is the capitalization mode. This is a special feature found only on the TSC Text Processing System and allows an upper case only terminal to be used for preparation of text which will later be output on a hardcopy device having lower case capabilities. The commands .CP and .NC turn this feature on and off respectively. If this mode is on, all letters are automatically converted to lower case unless preceded by a '@'. The '@' should be thought of as a typewriter shift key in its function. Another feature also enabled in this mode is similar to the 'shift and lock' on a typewriter. By typing a 'T' all characters following will be upper case until another 'T' is encountered.

It is often desirable, for readability, to use multiple spacing between lines. The TSC processor will allow this using the command .MS N where N is the space count desired and defaults to double spacing (N=2) if no value for N is given. The single space mode can be restored by either using .MS 1 or .SS for 'single space'.

Another group of commands deal with left margins and indentation. The left margin is normally set to 0 since the output device usually provides its own left margin (determined by paper positioning). Some applications require a wider margin at which time .LM N may be used to redefine it to be N spaces wide. Indent is similar to the left margin control with one difference. .LM N preserves the line length and simply moves the line to the right N spaces. .IN N, on the other hand, effectively reduces the line length by N columns in order to preserve the right hand margin. Setting the indent back to 0 will restore the full line length. Another form of indenting can be done by the use of the single indent command .SI N. Single indent is identical to indent except it is automatically restored to 0 after the line is output. It should be noted that the commands for left margin, indent, and single indent are additive in that if the following string of commands is issued:

```
.LM 10
.IN 8
.SI 5
```

the resultant output line would be preceded by 23 spaces, succeeding lines are preceded by 18 spaces assuming another .SI command was not used.

A note of caution is necessary concerning a characteristic of several of the processor's commands. Most commands will perform only their specified function but some also cause a line 'break'. A break is the forcing of output of the line currently being collected in the line buffer. Normally a line is kept in the buffer until the specified line length has been reached, at which time justification may or may not occur, depending on the mode enabled (also assuming that 'fill' is turned on). The break will cause the partial line to be output without being filled,

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

but the appropriate justification will be performed. This is useful for starting new paragraphs or new blocks of text. Some of the commands which cause a break are .CE, .FI, .NF, .IN, and .SI. Sometimes it is desirable that these commands do not cause a break. This can be done by using the 'no break' control character, ':'. So far, all commands have been shown preceded by the normal control character, a period. To set an indent of 10 and not cause a break, the following should be used:

:IN 10

The colon may be used with any command, whether the command normally causes a break or not.

It is often necessary to produce a section of one or more blank lines. The space command, .SP N, can be used to output N blank lines. The space command also causes a break. If N is not specified, the processor will output 1 blank line. It may be required that the blank lines all be on the same page, maybe for later insertion of a photograph or illustration. The TSC Text Processor allows this by using the 'save space' command, .SV N, where N is the number of lines required. If there are not N lines remaining on the current page, no line is output but instead, printing continues and the count (N) is saved for later use. When the next page is reached, the 'output saved space' command may be used, .OS, which will then produce the remembered number of blank lines. A convenient method for using .OS will be given later. Another similar command is the 'need lines' command, .NL N, where N is a line count. This command says that there must be N lines remaining on the current page, and if there are not, eject to the next page. This is convenient for keeping special blocks of text together (keep them from being split by page boundaries), or for not starting a new paragraph at the bottom of a page if only 1 or 2 lines will fit.

The commands which have been described so far will allow very nice page formatting. If these were all that were available in a text processor, much time and effort would be saved. The TSC Text Processing System, however, offers many more commands and much more versatility. One feature often needed in documents or manuals is the page title. There are many different ways of providing titles but the TSC processor uses a title command which has the form:

.TL 'field1' 'field2' 'field3'

where field1 is left-adjusted, field2 is centered, and field3 is right-adjusted. Any one or all of the fields may be present. Another feature supported in the TSC processor is the ability to print the current page number in the text. Any place a percent sign (%) appears, it will be replaced by the page number. A few examples will clarify the use of the title command.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

```
. TL 'Main Title'  
. TL ''Centered Title'Date'  
. TL '--%'
```

The first line will left adjust "Main Title" on the page. The second example causes "Centered Title" to be centered and "Date" to right adjusted. The final example will cause the current page number to be printed between two dashes.

Now it would be nice if there was some way of getting the title (and maybe a few other commands) to automatically execute at the top and/or bottom of each page of output. The TSC processor offers two advanced features to perform this task: macros and traps. A macro is a set of commands grouped together and given a name. When this name is later referenced, the entire group of commands will be executed. Essentially, what results is the ability to write special programs using the command set of the processor to do specific tasks such as headers, paragraphs, special titles, etc. The trap allows the user to specify a certain line on the output page where a specific macro is to be executed. To solve the title problem stated above it is convenient to define two macros, a header macro and a footer macro. The purpose of the header is to perform a sequence of commands to make the top of each new page appear the same. The footer macro works at the bottom of each page. Suppose it was required that the top of each page have three blank lines followed by a centered title and the bottom of each is to contain a centered page number between dashes. The following macros and trap placement would satisfy this requirement.

```
. DM HD  
:SP 3  
. TL ''Page Title''  
:SP 3  
--  
. DM FT  
:SP 3  
. TL '--%'  
:PG  
--  
. AT 1 HD  
. AT -7 FT
```

The .DM command is used to define a macro and the first one listed in the example defines the header macro called HD. The header macro will space down 3 lines (without causing a break since the no break control character (':') was used), print a centered title, and finally print 3 more blank lines without causing a break. The last line of the header macro definition is '..' and is the command for terminating a macro definition. The second macro defined is FT and is used for the footer. Upon execution it will space down 3 lines (without a break), print a

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

centered page number, and eject to the next page. The .AT commands were used to set the trap locations. .AT 1 HD causes the header macro to be executed at line 1 of every new page while .AT -7 FT causes the footer macro FT to be executed at the 7th line from the bottom of each page. The ability to specify trap locations and define macros makes titles and footers extremely simple and efficient.

One of the important aspects of using a text processor is the ability to make a few minor command changes and greatly change the final copy. As an example, suppose at the last minute it was decided that it would look better if there were four blank lines at the top of each page rather than three. If the document were being prepared by hand it would be necessary to retype the entire work to obtain the extra space. Using a small text processor it would only be necessary to go back and change the line count before each title. The TSC Text Processor and its ability to define macros means only one line in the entire text file needs to be changed. The second line of the header macro which is currently ':SP 3' would be changed to read ':SP 4!' One simple change and the desired result is obtained! It should be kept in mind that when preparing documents with a processor supporting macro capability, all of the often-used command strings should be defined in a macro so simple global changes may be easily performed if so desired.

There are more advanced features supported in the TSC Text Processing System. One of these is the ability to do conditional command execution. There are four forms of this command:

```
.IF O .XX  
.IF E .XX  
.IF N .XX  
.IF !N .XX
```

where O and E stand for Odd and Even page number respectively, and N can be a number, a number register (to be explained shortly), or an expression containing numbers and number registers. The exclamation point is the 'NOT' operator and .XX is any command or macro name. The command works as follows: IF the condition is true (page is odd or even, or the number or expression is greater than zero) the command 'XX' is executed, otherwise it is not. Preceding the expression by '!' will cause the command or macro to be executed only if the condition is not true (less than or equal to zero). The following special header macro definitions will illustrate the use of this command.

```
.DM HD  
:SP 3  
.IF O .TL '''Title'  
.IF E .TL 'Title'''  
:SP 2  
..
```

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

```
. DM HD
:SP 3
.IF %=1 .TL "Title"
:SP 2
```

The first header defined causes the title to be right-adjusted on odd numbered pages and left-adjusted on even pages. The second definition will print a centered title on each page except page number one since the value of the expression will be zero when the page number is one (remember that the '%' represents the current page number).

Another feature contained in the TSC processor is the ability to use number registers. Two types exist, one which allows the user to read and access certain system parameters including the date, page number, current indent, left margin, current column position, current line on the page, and line length. The second type are user definable and can be used exactly as variables would be used in a program. Number registers are the single letters A-Z and the percent sign (%) already introduced. Several other number register features are supported by the TSC processor, including auto increment, assigning values to the registers, use in expressions (as seen in the .IF command), and the ability to print any register on the output in either Arabic, capital Roman, or small Roman numerals.

Some processors, including TSC's, allow communication between the processor and the operator during actual text processing. Three of these commands take on the following form:

```
.ST
.TM any string
.GI any string
```

The first command will stop the processing and print 'STOP' on the user's terminal. This may be desirable if special paper positioning is required or other special action is needed. When the processor has been stopped it may be restarted by typing any character on the terminal except an 'S' which will halt processing. The second command listed will send 'any string' to the terminal as a special message. It may be used before the 'STOP' command to issue special instructions to the operator. The last command will 'Get Input' from the terminal and insert it into the output stream. 'Any string' can be used for a prompt. An example where this command is quite useful is in the preparation of form letters. The processor may prompt the operator for names and addresses which are then typed in at the terminal and automatically inserted into the text!

One final command will be described in this introduction, the 'divert text' command. Sometimes it is desirable to save text currently encountered for later use. An example of this is when trying to do footnotes. It would be nice if immediately

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

after the footnote reference was made, the actual footnote text could be typed, but saved for later insertion at the bottom of the page. The mechanism which allows this sort of operation is called a 'diversion' and is only available on the more complex text processors, such as TSC's. Two forms of the diversion usually exist:

.DI XX
.DA XX

where .DI instructs the processor to divert the following text into a diversion space named XX and .DA says to divert and append to the diversion space named XX. During diversion, all normal text processing still takes place, but rather than outputting the text to the printer, the text is written to a special place internal to the processor. The diversion process continues until the command for a divert is found without a name specifier. To recall the diverted text, it is only necessary to call it by name, exactly as macro calls are performed.

As an advanced exercise and demonstration of the diversion process (as well as many other processor commands) a complete set of macros for handling footnotes will be described. The reader should note that the following example is very complex and several readings will probably be required in order to fully understand its operation.

```
.NR B ?  
.DM HD  
:SP 2  
.IF %-1 .TL 'FOOTNOTE TEST'//  
:SP 2  
.AU 1  
.NR X 0  

```

- continued on next page -

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

```
DM EF
.BR
.EV 0
.DI
.NR W -#V
.CH FO #W
.IF #N-#P-#W ,CH FO #N+1

.DM SR
-----
.BR
.DM TR
.BF
.NF
.FE
.FI
.EF

.DM FN
.DI FE

.DM FT
.EV 1
.NF
.TX
.RM TX
.DI
.FI
.EV 0

.AT 1 HD
.AT -#B FO
.AT -4 NM
.CH FO 70
.AT -#B FN
.CH FO -#B
.EV 1
.AU 1
.LN 55
.EV 0
```

This example is quite similar to the one given in the "NROFF Users' Manual" written by J. Ossanna, of Bell Laboratories. To use these macros, merely insert their definitions at the beginning of the text file, and immediately after a footnote reference has been made, call macro BF. Following the call, simply type the footnote text and end it with a call to EF.

A description of the macros follows. The first line defines number register B and sets it equal to 7. Number register B is used to specify the size (in lines) of the bottom margin. A header macro definition follows (HD) and provides several functions. After spacing down two lines, the title is output

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

unless it is page number one (the IF command). Two more lines are produced and the auto increment value is set to one. Number register X is cleared and it is later used to keep track of the number of footnotes on the current page. Next, W is set to the location of the bottom margin trap and will later be adjusted as necessary if footnotes are added. The IF #V command checks to see if there is any remaining footnote text from the previous page and if so they are reprocessed (number register V contains the line count of the last diversion). Finally, the 'no space' mode is turned on to suppress any spaces which might otherwise get printed needlessly at the top of the page.

The footer macro, F0, clears the diversion count, V, and checks the value of X. If X is not zero (meaning there were footnotes on the page), macro FT is invoked. The footer is then restored to its original location by using the Change command as defined by B. The last command does a page eject. Macro NM is used. to print a centered page number at the bottom of each page.

The begin footnote macro, BF, starts with a divert append into the diversion space called TX. The environment* is switched, and if it is the first footnote on the page, macro SA is invoked which outputs a set of dashes as a simple footnote separator line. Diversion of the footnote text continues until macro EF is called. At this time a 'break' is executed and the original environment is restored. The diversion is stopped with the DI command. Number register W is updated by the number of diverted lines and the footer trap line is changed to compensate for the added footnote lines. Finally, if the number of diverted lines was great enough to move the footer trap up past the current line position, the trap is reset to the next line. TR is responsible for rediverting any lines of footnote text which will not fit on the page. It is very unusual for this to happen but this may occur if a footnote is very long and is referenced near the bottom of the page.

Macro FT is used for reading back the diverted text. It switches environments, sets the no fill mode, and calls TX, the actual footnote text. TX is then removed from the macro list, the fill mode is restored, and the environment switched. The last group of lines is used to define the trap locations of the various macros. The header is set to line one, and NM is set to execute four lines from the bottom of the page. The trap for the footer is planted at -#B, then moved past the bottom of the page while FN is also placed at -#B. F0 is then moved back as originally placed so in effect both F0 and FN are placed at the same line, but trap FN can only occur if the footer trap is moved up by the occurrence of a footnote. The last lines switch to environment one and initialize it for a line length of 55 and auto increment of one.

*Environment switching is a feature supported by many of the larger text processors (including TSC's) which allows all of the major environment parameters to change simultaneously.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

As a final example of how a text processor can be used, a sample section of text will be given. The text is shown first with the commands and then as the text processor would output the final copy.

```
. SP 2
. CE 2
↑TEST OF SEVERAL↑
↑PROCESSOR COMMANDS↑
. SP
. SI 5
@THIS EXAMPLE SHOWS HOW COMMANDS AND TEXT CAN BE INTERMIXED
FOR LATER PROCESSING BY A TEXT PROCESSOR.
@THE EXAMPLE STARTED BY CENTERING TWO LINES FOLLOWED
BY A SINGLE INDENT TO SIGNIFY THE START OF A PARAGRAPH.
@THE CAPITALIZATION MODE IS ON AND THE UPPER CASE SHIFT
CHARACTERS ARE BEING USED.
. SP
. LM 10
. LN 45
. JU C
@THE ADJUST MODE WAS JUST CHANGED TO CENTERING
AS WELL AS A LINE LENGTH OF 45.
@THE LEFT MARGIN WAS SET TO 10 TO GIVE A NICELY
CENTERED NARROW LINE.
@SPECIAL EFFECTS LIKE THESE ARE EASILY ACCOMPLISHED.
. SP
. LM 0
. LN 65
. JU N
@THE PARAMETERS WERE JUST SWITCHED BACK SO THE
LINE APPEARANCE WILL BE RESTORED.
THIS IS A SHORT EXAMPLE BUT SHOULD SHOW HOW THE
COMMANDS CAN BE INTEGRATED WITH THE TEXT.
```

This example appears in its expanded form on the next page.

This introduction to text processing is intended to be only that and is not a complete treatment of the subject. Many commands and features have been omitted. The ones included are the most general and the most used commands which offer the user a great deal of control and flexibility. Hopefully some eyes have been opened to the wide variety of applications of the text processor.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

EXPANDED EXAMPLE

TEST OF SEVERAL
PROCESSOR COMMANDS

This example shows how commands and text can be intermixed for later processing by a text processor. The example started by centering two lines followed by a single indent to signify the start of a paragraph. The capitalization mode is on and the upper case shift characters are being used.

The adjust mode was just changed to centering as well as a line length of 45. The left margin was set to 10 to give a nicely centered narrow line. Special effects like these are easily accomplished.

The parameters were just switched back so the line appearance will be restored. This is a short example but should show how the commands can be integrated with the text.

*NOTE: This entire user's manual was prepared using the TSC Text Editing System and the TSC Text Processing System.

Command Summary

~~buffer~~
line

Command Form	Initial Value	Default Argument	Cause Break*	Explanation
--------------	---------------	------------------	--------------	-------------

I. PAGE CONTROL

.PL +N	66 lines	66 lines	no	Page length.
.PG +N	N=1	-	yes	Eject to next page.
.PN +N	N=1	ignored	no	Page number.
.LM +N	N=0	previous	no	Left margin.
.NL N	-	N=1	no	Need N lines.

II. TEXT FILLING, ADJUSTING, AND CENTERING

.BR	-	-	yes	Break buffer.
.FI	fill	-	yes	Fill output lines.
.NF	fill	-	yes	No fill or justification.
.JU C/F	Jst, norm	just.	no	Justify on.
.NJ	just	-	no	No justification.
.CE +N	off	N=1	yes	Center N input lines.

III. VERTICAL SPACING

.MS N	prev	N=2	no	Multiple spacing.
.SS	single	-	no	Single space lines.
.SP N	-	N=1	yes	Space N lines.
.SV N	-	N=1	no	Save N lines.
.OS	-	-	no	Output saved lines.
.NSM B	space	-	no	No space mode on/off
.RS	-	-	no	Restore spacing.

IV. LINE LENGTH AND INDENTING

.LN +N	65	prev	no	Line length.
.IN +N	N=0	prev	yes	Indent.
.SI +N	-	N=1	yes	Single indent.
.PI ST	-	-	yes	Put string in indent.

V. MACROS, DIVERSIONS, AND LINE TRAPS

.DM XX	-	ignored	no	Define or redefine a macro.
.AM XX	-	ignored	no	Append to a macro.
.RM XX	-	ignored	no	Remove macro or diversion.
.DI XX	-	end	no	Divert out to macro "XX".
.DA XX	-	end	no	Divert and append to "XX".
.AT -N XX	-	-	no	Set trap at line N.
.CH -N -M	-	-	no	Change trap location.
.CH XX -M	-	-	no	" " "
..	-	-	no	End macro specification.

*The use of `:' as the control character (instead of `.') suppresses the break function.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

Command Form	Initial Value	Default Argument	Cause Break	Explanation
--------------	---------------	------------------	-------------	-------------

VI. NUMBER REGISTERS

.NR X +N	-	-	no	Number register.
.AU +N	0	prev	no	Set auto increment.
.AR	arabic	-	no	Arabic numbers
.CR	arabic	-	no	Capital Roman numbers.
.SR	arabic	-	no	Small Roman numbers.

VII. TABS AND TAB CHARACTERS

.TR N...	none	none	no	Set tab columns.
.TF C	un sp. *	un. sp. *	no	Set tab fill character.
.TC C	none	none	no	Set tab character.

VIII. THREE PART TITLES

.TL 'left'center'right'		no	Define title.	
.LT +N	65	prev	no	Length of title.

IX. CONDITIONAL INPUT COMMANDS

.IF C COMMAND	-	no	If true, do command.
.IF !C COMMAND	-	no	"
.IF N COMMAND	-	no	"
.IF !N COMMAND	-	no	"

X. ENVIRONMENT SWITCHING

.EV N	N=0	N=0	no	Change environments.
-------	-----	-----	----	----------------------

XI. SPECIAL CONTROL COMMANDS

.CP	no caps	-	no	Capitals mode on.
.NC	no caps	-	no	No caps mode.
.ST	-	-	yes	Stop processing.
.EX	-	-	yes	Exit processor.
.PS	no pass	-	no	Pass text without proc.
.RP	-	-	yes	Repeat entire file.
.DH	-	-	yes	Double height line**.
.DW	-	-	yes	Double width line**.
.DB	-	-	yes	Double height and width**.

E = English T for type
F for type

*Un. sp. = unpaddable space character.

**These commands require the output device to support double dimensioned character printing.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

Command Form	Initial Value	Default Argument	Cause Break	Explanation
--------------	---------------	------------------	-------------	-------------

XII. EXTERNAL COMMUNICATION

.TM ST	-	-	no	Send string to terminal.
.GI ST	-	-	no	Get line from terminal.

XIII. MISCELLANEOUS

*	-	-	no	Comment field.
---	---	---	----	----------------

SPECIAL CHARACTER DEFINITIONS

Character Meaning

\	Standard escape character.
@	Force capital letter.
↑	Set capital letter mode.
#	Number register specifier.
.	Basic control character.
:	No break control character. <i>(immediate execution)</i> <i>buffer 1000</i>

NUMBER REGISTERS

Register Meaning

A-B	User definable
C	Current column count
D	Day of the month
E-F	User definable
G	Get Input (. GI) character count
H	User def.
I	Current indent
J-K	User def.
L	Current line length
M	Month
N	Line count on page
O	Current left margin
P	Current page length
Q-U	User def.
V	Last diversion line count
W-X	User def.
Y	Year (2 digits)
Z	User def.
%	Page number.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

Reference Manual

INTRODUCTION

All input lines to the processor which are to be interpreted as commands should be started with the control character (a `.' or `:'.) in column one followed immediately by the two letter command. If the characters are not system command names or user defined macros, the line will be ignored. The 'nobreak' control character (':.') may be used with any command to suppress normal line breakage during processing. Only a single command reference is permitted on any one line.

The following detailed command descriptions reference numerical arguments either as N, +N, or -N. N means any argument is taken as absolute and any previous value is simply replaced by the new value. +N is used when the argument may take any form of a number (either positive, negative, or absolute). Valid arguments of this form are +4, -10, and 3 where the old value would be incremented by 4, decremented by 10, and replaced by 3 respectively. Arguments of the form -N may use absolute values or negative values which are subtracted from the current page length (to reference N number of lines from the bottom of the page). When expressions are involved using the +N argument, the entire N is evaluated before the increment or decrement is applied (e.g. -6-3 will decrement the value by 3). Certain commands requiring arguments will keep the last argument assigned if the argument field is left empty when the command is called.

I. PAGE CONTROL

The page control commands are used to set the physical page parameters such as length, width, margins, numbering, etc. Top and bottom margins are not automatically provided and should be defined by the user with macros as described in a later section.

- .PL +N Set page length to N lines. Initial value is 66 lines and is reset to 66 if no argument is given. Does not cause a break. The maximum N is 255.
- .PG +N Eject to next page. If N is given the new page number will be adjusted accordingly. The page number is automatically incremented if no argument is given and the command does cause a break. Max N is 255.
- .PN +N Set the page number to +N. If .PN occurs before the first break or first text, it will be set for the first page. The value is initially 1 and the command does not cause a break. The maximum page number is 255.
- .LM +N Set the left margin according to +N. The entire output line will be offset to the right by the number of spaces the current LM is defined. Initially there is

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

no margin ($N=0$) and no break occurs. Left margins should not exceed 100.

- .NL N Need N lines on the page. If the distance to the next trap position or the bottom of the page is less than N, the paper is advanced to the next trap position (blank lines output). Otherwise no action takes place. No break occurs and the default argument is $N=1$.

II. TEXT FILLING, ADJUSTING, AND CENTERING

The following commands affect the appearance of individual lines of text. Two important parameters are referenced, Fill and Justify. The default fill mode is to fill output lines with as many words as possible without exceeding the set line length value. Any extra words are saved for output on the next line. A word is defined to be any string of characters separated by a space or spaces. If two words are to be separated by a space but are not to be split across line boundaries or separated by the justification routines, the unpaddable space character, "\ " (slash space) may be used. The default justification mode is left and right, giving straight margins on both sides. Filled lines which contain too few character positions to completely fill-out the specified line length are padded with spaces until the correct length is achieved. The space filling or padding is done from alternate sides of the page as each line is justified to eliminate 'white rivers' which may otherwise occur in the text. No hyphenation is performed. It is important to note that fill must be on in order for the justification to be performed, but fill may be on by itself. If fill mode is off, characters are passed exactly as they appear on the input file.

- .BR Break the line currently being filled in the buffer. The line is output after specified justification is done but no further filling or padding is attempted. Input lines beginning with spaces and empty text lines (blank lines) also cause a break.

- .FI T Fill mode is turned on and subsequent output lines are filled. This command causes a break.

- ~~.NF~~, FI F Turn off fill mode (nofill). Following input lines are neither filled or justified, but are copied to the output exactly as they appear on input, without regard to the current line length. Causes a line break.

- .JU C Justification is enabled. If fill mode is off, adjusting will be deferred until it is back on. If the justify type character, "C", is present the justification type is set as follows: N sets for normal (default, left and right), R sets right only justify, and C will center lines (both margins ragged). If the type character is absent, justification is turned back on with the type previously used. No break is caused.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

. NJ JF Turn justification off. If fill is on, the resultant output line will have a straight left and a ragged right edge. No break is caused and the justify type remains unchanged.

. CE +N Center the next N input lines. A break occurs before the command and then automatically after each line is output. If the resultant line is longer than the current line length, the output line will be left hand adjusted. The maximum count is 255.

III. VERTICAL SPACING

All line spacing defaults to standard single spacing. It may be set at any time by using the MS command. If the line spacing is N, N-1 blank lines are inserted after each output line. The occurrence of a trap will terminate any remaining spacing count. Contiguous space should be saved by using the SV and OS commands.

. MS N Set multiple line spacing to N. N-1 blank lines are inserted after each output line. No break is caused and if N is not specified the value of 2 will be used (double spacing). Max value is 255.

. SS Set single space mode. No blank lines are output after text lines and no break occurs.

. SP N Space N lines. The number of output lines is limited to the distance to the nearest trap or bottom of the page. If nospace mode is on, no spaces are output. If no value for N is given, it defaults to 1. SP causes a break.

. SV N Save N lines of space. If the distance to the next trap (or the bottom of the page) is greater than N, N lines are output, otherwise no lines are immediately output but the count (N) is saved for later output (see OS). Subsequent SV commands will overwrite any previously remembered N. Nospace mode has no effect. The command does not cause a break and the default value for N is 1.

. OS Output saved space. This command is used to output any previously saved space from the SV request. The remembered count is cleared after calling OS and nospace mode has no effect. A break does not occur.

. NS SP C ~~on/off~~ Nospace mode is turned ~~on~~. The nospace mode inhibits SP requests and PG requests without a next page number. This mode is automatically turned ~~off~~ after the output of a line of text. No break is caused.

. RS RH Restore space mode. If the nospace mode is ~~on~~, it is turned off with this command without causing a break.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

IV. LINE LENGTH AND INDENTING

Using the following set of commands, the user has complete control over the line length and various forms of indenting. The line length includes all indent spaces but does not include left margin spacing. As long as the fill mode is turned on, the resultant output line will be less than or equal to the current line length minus the indent. Line lengths of less than 6 columns are not permitted.

- .LN +N Set line length. The initial value is 65 columns and the command does not cause a line break. Line lengths must be between 6 and 255 columns inclusive.
- .IN +N Set the line indent according to N. With a line length of L and an indent of N, N spaces are output before each line and the remaining text is restricted to a size of L-N. Initially the indent is 0 and the command causes a break.
- .SI +N Single indent N spaces. Only the next output line will be indented by the amount specified by N. Note that single indenting may be done backwards into an indent field. (e.g. if indent is 10, SI -4 would temporarily set the overall indent to 10-4 or 6). IN and SI counts are cumulative and the final value may not be negative! This command causes a line break.
- .PI ST Put string in indent field. The string represented by "ST" (leading spaces ignored), is inserted into the field normally filled with spaces by the indent count. If the string is longer than the indent count, the string will be truncated so it will not extend past the indent field.

V. MACROS, DIVERSIONS, AND LINE TRAPS

A macro is a set of commands and/or text which can be assigned a name and called by name at a later time. All macro names are two characters long and must be different from any names already in existence in the system command name table. Macros are defined or redefined by using the DM command, or by using the output diversion command, DI. Macros already in existence may be appended to by using the RM or DA commands. If a macro is named XX, it may be invoked by an input line beginning with ".XX". A trap may also be placed at a specific vertical page placement to cause automatic macro execution at that point by using the AT command. During macro definition, number registers are not expanded into numeric values but are at the time the macro is executed. No other special character translation is done during macro definitions (e.g. tab expansion, etc.). Keep in mind that macros may be any combination of commands, macro calls, and text, but a macro may not define another macro (it may create a diversion).

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

A diversion is treated as a macro upon execution but is created in a different manner. Processed output may be diverted into a macro space for such purposes as footnote processing or vertical page size determination for conditional changing of page parameters (number register V contains the last diversion line count). All normal processing takes place during a diversion except left margins. It is standard practice to read back the diverted text in 'nofill' mode to suppress further line processing.

If at any time during macro definitions or diversion creation the macro space is overflowed, a system error will be generated and processing will be halted. None of the macro commands cause breaks in the line filling.

- .DM XX Define or redefine a macro with the character name XX. The actual macro begins with the next input line. The macro definition is copied until the termination character ".." is found starting in column 1. Macros may not contain DM requests but may create diversions.
- .AM XX Append to the macro named XX. This command acts exactly like DM except the following input lines are appended to an existing macro rather than creating a new named space.
- .RM XX Remove macro or diversion. The macro named XX is removed from the name list and subsequent calls to this name will have no effect.
- .DI XX Divert output into the macro space named XX. The macro named XX is defined or redefined at this point. All normal text processing occurs during diversions except left margin page offsetting is not done. The diversion process is ended when another DI or DA is encountered. Diversions can not be nested! The count of the number of lines last diverted is kept in number register V for possible later reference.
- .DA XX Divert append version of DI. The same rules apply for both commands.
- .RT -N XX At line N invoke macro XX. Any macro previously planted at line -N is replaced by XX. N is measured from the top of the page (0 or 1 may be used to represent the top) and -N is measured from the bottom of the page (e.g. if the page length is 66, line -1 represents line 66). If no macro name is given with the command, the trap located at line -N, if any, is removed.
- .CH -N -M Change trap. See next.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

.CH XX -M Change the trap planted at line -N to occur instead at line -M. Alternately, change the location of the trap for macro XX to line -M. If there is not a trap set at -N, the request is ignored.

Terminate a macro definition.

VI. NUMBER REGISTERS

Number registers are a type of variable used during processing. There are two classifications, user definable and system. Number registers have single character names (A through Z and '%'). Number registers may be used any time a number is expected in a command and also may appear imbedded in text. There are two methods of referencing a number register:

#X
#+X

where '#' is the register designator character and X is the name of the register. When using '%' it should not be preceded by the '#'. The '+' in the second example specifies that the number register is to be auto incremented prior to its use and it will retain the new incremented value. The auto increment amount is set using the AU command. When a number register reference is encountered it is converted to decimal, lower case Roman, or upper case Roman, as determined by the mode set. Number registers appearing in macro definitions are not converted until the macro is actually executed. Number registers may also be used to construct expressions any time a number is expected in a command (expressions may not be imbedded in text). The expressions are evaluated left to right and may contain only the operators '+' and '-'.

.NR X +N Assign a value to number register X. This command should only be used to assign values to user definable number registers.

.AU +N Set the auto increment amount to +N. Any time a register is referenced as "#+X", the AU value will be added to it prior to its actual use.

.AR Arabic numbers. See below.

.CR Capital (upper case) Roman numbers. See below.

.SR Small (lower case) Roman numbers. Number registers will subsequently be converted into Arabic, capital Roman, or small Roman respectively. This mode is initially Arabic and also applies to the outputting of page numbers using the '%'.
-22-

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

The following is a list of the system and user definable number register names

Register Meaning

A-B	User definable
C	Current column count
D	Day of the month
E-F	User def.
G	Get input (.GI) character count
H	User def.
I	Current indent
J-K	User def.
L	Current line length
M	Month
N	Line count on page
O	Current left margin
P	Current page length
Q-U	User def.
V	Last diversion line count
W-X	User def.
Y	Year (2 digits)
Z	User def.
Z	Page number

VII. TABS AND TAB CHARACTERS

The currently defined horizontal tab character is replaced by the required number of fill characters corresponding to the distance to the next defined tab stop column (on the line currently being filled). The fill character is normally the unpaddable space character but may be defined by using the TF command. Up to 20 tab stops may be defined and should be set in ascending order. Initially no tab stops are defined and the tab character is null. Any non alphanumeric character may be defined as the tab character. It should be noted that using tabs with the fill mode turned on can result in nonsensical output tab fields since the user may not know what the current output column is.

- .TA N... Tab stop settings. The default tab stops are all null (none) and a total of 20 may be defined. The stop values may be separated by spaces, commas, or any other nonnumerics, e.g. TA 10,20,25,40.
- .TF C Set the tab fill character. This is normally the unpaddable space character but may be defined to any nonnumeric printable character. If 'C' is not specified the fill defaults to the unpaddable space character.
- .TC C Define the tab character. Initially the tab character is null (none) but may be defined to any nonnumeric printable character. If 'C' is not specified the tab character again becomes null.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

VIII. THREE PART TITLES

Very convenient titling may be performed by using the TL command. Three fields may be used for left, centered, and right justification of titles. All 3 fields may be used or any combination of fields. The justification is done with respect to the title length which is independent of the defined line length. This length is initially 65 columns. The use of TL has no effect on current line accumulation (does not cause a break). TL is usually used in header and footer macros. For example, .TL '~-%' will print the page number in the center of the title length.

.TL 'LEFT'CENTER'RIGHT'

Place titles adjusted according to field. The strings represented by "LEFT", "CENTER", and "RIGHT" are respectively left adjusted, centered, and right adjusted within the current title length. Any of the fields may be empty and any nonnumeric printing character may be used in place of the field delimiter "''. The "%" character will be replaced by the current page number in Arabic or Roman representation.

.LT +N Set title length. The lengths of titles and lines are separate parameters. Indents do not apply to titles but left margin adjustment does.

IX. CONDITIONAL INPUT COMMANDS

Input command and macro calls may be performed on a conditional bases. Chained conditionals are permitted as in: IF #A IF #B .XX.

.IF C COMMAND See next

.IF !C COMMAND "

.IF N COMMAND "

.IF !N COMMAND

IF is the conditional command. "COMMAND" can be any system command or macro name. "C" is a built in condition code and can be either O or E to represent Odd or Even page numbers respectively. "N" is any number and can be a number, a number register, or any combination of these in the form of an expression using addition and subtraction. If the condition is true (the built in condition is satisfied or the number is greater than zero), the command or macro named is executed, otherwise the command is ignored. If "C" or "N" are preceeded by a '!' (not), the command is executed if the condition is false or the number is less than or equal to zero.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

X. ENVIRONMENT SWITCHING

There are a number of parameters which control the text processing and are grouped together and called the environment. These environment parameters may be changed all at once using the switch command. There are two environments, 0 and 1. They both have identical initial values for all parameters. Parameters within these environments are those associated with:

line length	vertical line spacing
indenting	centering count
adjusting	auto increment
filling	partially collected words
title length	partially collected lines

All other parameters are global, or in other words, they are not switched with the environment but remain unchanged. Examples of global values include left margin, page number, current line number, number registers, trap tables, and macro definitions. Since partially collected words and lines are kept with the environment, switching environments will not cause a break and will also preserve any left over words.

.EV N Change to environment N where N can be 0 or 1. If N is left null, environment 0 is assumed

XI. SPECIAL CONTROL COMMANDS

The following commands control certain aspects of the processor. The double height and width commands are hardware dependent. You should refer to the "adaption" section of this manual for details.

.CP Turn capital letter mode on. When enabled, this mode will allow the use of an upper case only terminal to prepare text for later output to a device which supports both upper and lower case. Each character is automatically converted to lower case unless it is immediately preceded by a '@' at which time that character remains upper case. Strings of characters may be kept in upper case by enclosing them between up arrows "↑". The "@" is like a shift key and the "↑" acts like a shift and lock key.

.NC Turn off capitals mode. Initially this mode is off and the special capitalization characters ("@" and "↑") are ignored.

.ST Stop causes processing to temporarily halt and the word "STOP" is output to the terminal. At this time, typing an "S" will cause all processing to be stopped and the processor will be exited. Typing any other character will cause processing to continue. The stop command does cause a line break.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

- . EX Exit the processor. Text processing is stopped just as if all input had been finished. This command is useful in conjunction with the IF command.
- . PS Pass all input to the output. This command is primarily intended as a debugging aid since it allows all input (including command lines) to be passed to the output. No command interpretation or processing is done and once in this mode, the remaining text will be passed until the end of the input file is reached.
- . RP Repeat processing on file. This command will cause the file to be 'rewound' and all processing to be repeated. This is useful for some form letter type applications.
- . DH Print the next line in double height characters. This feature requires special hardware on the output device. Consult "Adaptions" for details.
- . DW Print the next line in double width characters. Requires special hardware.
- . DB Print next line in both double height and double width characters. Requires special hardware.

XII. EXTERNAL COMMUNICATION

Two commands exist which allow for communication between the processor and the user during actual text processing. The TM command is useful for sending special instructions to the terminal such as paper adjustment or character font change information. The GI command can be used in form letter preparation or insertion of special text strings while processing is taking place.

- . TM ST Send a message to the terminal. ST may be any string of characters or words. The leading blanks are ignored. The message is simply output to the terminal and may be used before the Stop command to issue special instructions.
- . GI ST Get input from the terminal. If ST is present (any string), it is output to the terminal as a prompt message. Characters typed from the terminal following the execution of GI are automatically inserted into the input stream for text processing. This command can be used to get name and address information for form letter preparation. The 'get input' function is terminated by typing a carriage return; therefore, only one line of text may be entered with each GI command executed. After completion of the command, the number register G contains the character count of the string typed (not including the carriage return).

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

XIII. MISCELLANEOUS

The following describe some of the smaller features of the text processor.

- * Comment field. This may be used to insert comments into the input text and will be ignored by the processor. No output is created with this command (the comment is not passed to the output).

Special Characters

- \ Standard escape character. This character is used to remove special meaning from a character. For example, if a percent sign ("%) is needed in the output it is necessary to precede it with the "\", otherwise it will be interpreted as the page number (e.g. \%). To print a backslash, "\\\" must be used.
- @ Force upper case letter if in the capitals mode (CP). This acts similar to the 'shift' key on a typewriter. Example: "@test" will be output with an upper case "T" and lower case "est".
- ↑ Upper case string delimiter. This character acts similar to the 'shift and lock' key on a typewriter. As an example, ↑this is a test↑ would cause "this is a test" to be output in all upper case characters. The capitals mode must be on (CP).
- # Number register specifier When an alphabetic character is immediately preceded by a "#" it will be interpreted as a number register. Example: "#A" refers to number register "A".
- .
- :
- % The period is the basic command control character. If in column one, it specifies a two character command or macro name follows
- :
- % The colon is the no-break control character. It functions exactly like the period, but will suppress breaks caused by various commands.
- % Page number symbol. Any place the percent sign appears, it will automatically be replaced by the current page number.

Special notes

- A. Any time input is being typed into the processor, typing a 'control X' will delete that line and issue a "?" as a prompt.
- B. The processor automatically makes sure there are two spaces after ".", "!", or "?". This does not apply to punctuation immediately followed by another character.

Using the Text Processor

I. BRINGING UP THE SYSTEM

It is assumed the text processor is resident in memory and that any necessary adaptions have been made. The entry point to the program is hex 200. The system should respond with the copyright message and the prompt:

DATE (MM:DD:YY)?

If the date is not required by your program, simply type a carriage return. If the date is required, it should be entered in the form shown. Spaces or nonalphanumeric characters may be used between entries. As an example, to enter June 20, 1977, type the following:

: 6 20 77

If an error is made during entry, simply type a 'control X' and a new question mark "?" prompt will be issued. At that time, re-enter the entire date. The 'control X' may be used any time the processor prompts for input. The next prompt issued is:

TYPE P FOR PRINTER...

Typing a P will route all processor output to the printer device (using the user supplied printer I/O routines - see "Adaptions"). Any other character will cause the output to be sent to the terminal. If the printer was not selected, the next prompt will be:

LINES PER SCREEN?

Typing a carriage return will result in uninterrupted output from the processor to the terminal. Typing a number will cause that number of lines to be output, at which time processing will temporarily halt. To restart, type any character except a carriage return and the processing will continue. A carriage return will cause the processor to be exited. The final prompt is:

PAGE LIMITS?

and is used to specify a particular block of pages to be processed. Typing a carriage return will cause all pages to be processed and output. Typing two numbers separated by a space or graphic character will cause only the pages between those numbers (inclusive) to be output. For example, typing:

10,16

will result in only pages numbered 10 through 16 to be output.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

If just one number is entered, the processor will start outputting at that page number and continue to the end of the file. It should be noted that the processor always starts numbering the first page as number one unless instructed otherwise. As the processor is working, it may be stopped at any time by pressing the "BREAK" key on the terminal. This key must be held down at the end of line output. The processor will respond with:

..BREAK..

output to the terminal. At this time processing may be continued by typing any character except an "S" which will cause the processor to be exited.

II. GENERAL USE

There are several things to keep in mind while preparing text for the text processor. Remember that all commands must begin in column one. It is usually most convenient to begin each sentence on a new line for easy future editing. Macros should be used as often as possible. The reason for this is to keep global changes as simple as possible, e.g. change only one line in a macro as opposed to changing single commands scattered throughout the file. It is not necessary to understand how the macros provided in this manual work in order to use them. All that is necessary is to know how to use them which is thoroughly explained. As experience is gained with the processor, you will be able to create your own special purpose macros for easy formatting.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

MACRO LIBRARY

The following macro descriptions range from simple header and footer macros to a very complex footnote macro. It is not necessary to understand how the macros work, just how to use them. Each macro includes a description of what it does and how it can be used.

I. HEADERS AND FOOTERS

These macros are used to define top and bottom margins and also specify the contents of these margins, such as page numbers, titles, etc. Almost all processing jobs will require some sort of header and footer. Usually the macro definitions are placed at the beginning of the file (they need to appear before they are called for execution). The "AT" command is used to set the trap location (the line at which the macro should automatically execute) of each of the macros. Headers are set to line 1 and footers to a specific distance from the bottom of the page. Once these macros have been defined and their trap locations set, they can be forgotten about since the processor will do all the rest of the work. The first macro is a simple header macro which provides two blank lines, a centered title, and two more blank lines at the top of each page.

```
.DM HD
:SP 2
.TL "CENTERED TITLE"
:SP 2
.NS
.OS
..
.AT 1 HD
```

All of the header macros will contain a NS and OS command. NS will suppress any unnecessary spacing which may occur due to the unpredicted appearance of a SP command. For example, if the start of a new paragraph just happens to start at the top of a new page, there is no reason for the paragraph macro to space down two lines, since we are at the top of the page. NS will keep this from happening. The OS command instructs the processor to output any 'saved space' from the previous page. The next header is a little fancier. It does everything the previous one does except the titling is done a little differently. Here, if the current page number is even, the title is left hand justified. If the page is odd, the title is right hand adjusted.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

```
. DM HD
:SP 2
. IF E . TL "EVEN TITLE"
. IF O . TL "ODD TITLE"
:SP 2
. NS
. OS
```

```
..
```

```
. AT 1 HD
```

Subtitles may be used by simply placing a second TL command which contains the subtitle. The last header example is for those using a printer which uses separate sheets of paper (as opposed to continuous fed). This macro will issue a message to the terminal which instructs the operator to insert a new sheet of paper, before each page of text is processed. The paper should be set up such that the first line printed will be the top line of the paper. The operator will have to type a character on the terminal after each stop to restart the processor. Remember that typing an "S" will halt the processor.

```
. DM HD
. TM INSERT NEW SHEET
:ST
:SP 2
. TL "TITLE"
:SP 2
. NS
. OS
```

```
..
```

```
. AT 1 HD
```

Footer macros are similar to headers...except they are set to execute at the bottom of a page. For example, specifying AT -6 F0 would cause the macro called F0 to automatically execute at the 6th line from the bottom of the page. The first footer gives a five line bottom margin with the page number between 2 dashes centered on the page, 3 lines from the bottom.

```
. DM F0
:SP 2
. TL "--%--"
:PG
```

```
..
```

```
. AT -5 F0
```

It is often desirable to have page numbers on every page except page number one. The following footer will do exactly that.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

```
. DM FO
:SP 2
:IF %-1 . TL 11-2-11
:PG
...
. AT -5 FO
```

There are several other types of header and footer macros which can be created. Some of these appear in the macros which follow.

II. PARAGRAPHS AND HEADINGS

There are many forms of paragraphing. The TSC Text Processor does not restrict one to using one particular form. One type of paragraph is to produce one blank line and start the first line of the paragraph indented five spaces. The following macro does just that:

```
. DM PP
. SP
. SI 5
...
```

To use the paragraph macro, simply call it by name any time a new paragraph is desired (e.g. type ".PP" in column one). One little feature which may be added to the macro is a need lines command, NL. In the following example, NL 3 is used to tell the processor that we desire at least three lines be left on the page before a new paragraph is started. This will keep one or two lone lines from being placed at the bottom of the page.

```
. DM PP
. SP
. NL 3
. SI 5
...
```

Many other types of paragraph macros may be created along the same lines as those presented.

Another useful macro can be created for major heading creation. One type of major heading might have a centered title spaced two lines down from the last line of text. The macro to accomplish this may look as follows:

```
. DM MH
. SP 2
. CE
...
```

To use this macro, type ".MH" when the heading is desired. The next line should contain the heading title. For example:

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

Line of text.

. MH

Heading Title

The last two macro examples are quite simple, but show how even two or three lines of commands may be replaced by a single macro call. This is quite useful if these operations are going to be repeated many times throughout a document.

III. FOOTNOTES

The following set of macros is all that is required to do very efficient and easy footnote handling. A description of how they actually work is contained in the introduction of this manual. To use these macros, it is only necessary to include their descriptions at the beginning of your file. As soon after a footnote is referenced in the text, call the macro BF (begin footnote) to begin the footnote. Immediately following this call, type the contents of the footnote, followed by a call to the macro EF (end footnote). The following serves as an example:

```
Text here referencing a footnote*
.BF
*Footnote contents typed here and
may be several lines long.
.EF
```

It should be noted that the footnote macros contain their own header and footer macros which may be modified as desired. These macros should be the first lines of a file.

```
. NR B 7
.DM HD
:SP 2
.IF %-1 .TL 'FOOTNOTE TEST'
:SP 2
.AU 1
.NR X 0
.NR W 0-#B
.IF #Y .TR
.NS
.
.DM FO
.NR Y 0
.IF #X .FT
.CH FO -#B
:PG
.
.DM NM
.TL '--%-'
.
.
- continued -
```

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

```
. DM BF
. DA TX
. EV 1
. IF !#+X-1 . SR

. DM EF
. BR
. EV 0
. DI
. NR W -#Y
. CH FO #W
. IF #N-#P-#W . CH FO #N+1

. DM SR
-----
. BR

. DM TR
. BF
. NF
. FE
. FI
. EF

. DM FN
. DI FE

. DM FT
. EV 1
. NF
. TX
. RM TX
. DI
. FI
. EV 0

. AT 1 HD
. AT -#B FO
. AT -4 NM
. CH FO 70
. AT -#B FN
. CH FO -#B
. EV 1
. AU 1
. LN 55
. EV 0
```

Please remember that it is not necessary to fully understand how these macros work as long as you know how to use them.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

IV. TWO COLUMN OUTPUT

The TSC processor does not support backward line feeds so it is necessary to use some operator intervention in order to produce two column output. The following set of macros will produce two column output, each column being 31 characters wide. When the text of the first column reaches the bottom of the page, the string "REPOSITION PAPER" will be output to the terminal and a "STOP" command is executed. At this time the operator should reposition the paper to the top of the page and then restart the processor by typing any key but "S".

```
. LN 31
. NR A 0
. DM HD
. IF #A PA
:SP 2
:
. AU 1
. IF !#+A-1 . TL //title//
. IF #A-1 :SP
:SP 2
. IF #A-1 . LM 34
:
. DM FO
:SP 2
. LM 0
. IF #A-1 . TL //-%-//
. IF #A-1 . NR A 0
:PG
:
. DM PR
. TM REPOSITION PAPER
:ST
. PN %-1
:
. RT 1 HD
. RT -5 FO
. BR
```

It should be noted that these macros also contain their own special set of header and footer macros which may be modified as desired.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

V. FORM LETTERS

The last set of macros and examples deal with form letters. These macros are shown with some sample text. The RP (repeat) command is used so that the file is repeated over and over, until just a carriage return is typed in response to the "NAME?" prompt, at which time processing halts. The macro creates a name and address header at the top of each page. Following is "Dear <persons name>" and the text of the letter.

```
. JU N
. SP 6
. NF
. DI NM
. GI NAME
. BR
. DI
. IF !#G . EX
. NM
. GI STREET
. GI CITY STATE & ZIP
. SP 3
. FI
DEAR
. NM
. SP
. SI 5
THANK YOU FOR YOUR INTEREST IN THIS ORGANIZATION.
WE ARE SURE THIS IS THE START OF A LONG PROFITABLE
FRIENDSHIP.
IF YOU HAVE ANY QUESTIONS, FEEL FREE TO CONTACT US.
. SP 2
. JU R
THANK YOU
. PG
```

More intricate form letter macros could certainly be created. This one shows how the 'name' can be saved for later use in the body of the letter.

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

System Adoptions

I. PROCESSOR INITIALIZATION

Before the processor is run, three pieces of information need to be supplied to the program

A. FSTRAM (loc. \$0097-0098): These two bytes should be set to contain the address of the first byte of source text in RAM. For example, if your text file begins in RAM at location \$3000 then FSTRAM (\$0097 & \$0098) should be set to \$3000.

B. LSTRAM (loc. \$0099-009A): These two bytes should be set to contain the address of the first empty byte past the source text file. For example, if your text file's last character (a carriage return) is at location \$3A50, then LSTRAM (\$0099 & \$009A) should be set to \$3A50+1 or \$3A51.

C. JNKCNT (loc. \$009D): This byte should be set to contain the number of bytes following the carriage return in each line before good text resumes. The processor needs text lines in the following form:

text text text C.R. text text C.R. etc.
where C.R. represents a carriage return (\$D). If there are any non text characters after the carriage return, the processor must be instructed of these so it can ignore them. For example, the TSC Text Editing System saves text in memory with 3 line number bytes preceding each line (after each carriage return). In this case, JNKCNT (\$009D) should be set to 3.

II. INCH

INCH is the routine called in MIKBUG for character input. If not using MIKBUG you will have to provide your own input routine. It must not destroy B or X and should return the ASCII character with parity removed in the A accumulator. This routine is called by a JMP instruction (7E) from location \$0206 and is presently set to \$E1AC.

✓ CØ!

III. DUTCH

DUTCH is the routine called to output a character to a terminal. It is presently referencing the output character routine in MIKBUG (\$E1D1). If you are supplying your own, it must not destroy B or X and should output the ASCII character contained in the A accumulator. This routine is called by a JMP instruction from location \$0203.

✓ CØ!

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

IV. MONITOR

MONITOR is the exit point from the processor. It currently is a JMP to \$E0D0 (MIKBUG). To set your own address, change the JMP instruction located at \$B289. FSC

V. STACK REFERENCES

The stack pointer is initialized to \$01FF at two different locations in the program. If you are moving the stack location, be sure to change both references. The stack is loaded at \$0212 and \$0CC5. Nog
TH

VI. PRINTER ROUTINES

There are two printer routines referenced by the processor, PRNIT (\$15A4) for printer port initialization and PROUCH (\$15C4) for output a character to the printer. These routines need to be supplied and are simply set up as RTS (39) in the processor. 32 bytes have been reserved for each routine. The PRNIT should do any necessary port and/or printer initializing. PROUCH should output the contents of A to the printer keeping B and X preserved. These routines are called from \$0200 and \$020F.

VII. TEST BREAK ROUTINE

The routine at location \$1471 is used to check for the occurrence of the "BREAK" key during processing. This routine assumes a PIA is being used for I/O in a MIKBUG configuration. If an ACIA port is being used, substitute the following code in for the TSTBRK routine. The processor can now be halted by typing a 'control C' during text processing.

*
* BREAK ROUTINE FOR ACIA
*

		ORG	\$1471	
1471	B6 80 04	TSTBRK	LDA A ACIA	ACIA BASE ADDRESS
1474	44		LSR A	GET STATUS
1475	25 01		BCS TSTBR4	
1477	39	TSTBR2	RTS	NO CHARACTER
1478	B6 80 05	TSTBR4	LDA A ACIA+1	GET CHARACTER
147B	81 03		CMP A #\$03	IS IT 'CTL C'?
147D	26 F8		BNE TSTBR2	
147F	CE 15 92		LDX #BRKSTR	POINT TO STRING
1482	7E 0C D8		JMP STOP1	OUTPUT IT
8004	ACIA	EQU	\$8004	SET AS REQUIRED!

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

VIII MACRO STORAGE SPACE

The macro storage space is presently set to approximately 1.5K and resides between \$19F4 and \$1FFE. If it is found that more space is required, the end pointer to the space (LMACRO) is referenced at \$02FE and may be changed as needed.

IX DOUBLE CHARACTERS *

Three commands exist in the processor which require special printer hardware. These are double height (DH), double width (DW), and double both (DB). Some commercially available printers will print single lines of double size characters if a special control character is received prior to the line. The double height control character (\$12) is referenced at location \$0C08. The double width control character (\$0E) is referenced at locations \$0C13 and \$0C1C. These may be changed as required.

X MAKING THE TSC TEXT EDITOR AND PROCESSOR CO-PRESIDENT.

Following is a description of the steps necessary to relocate the TSC TEXT EDITING SYSTEM, allowing co-resident operation with the TSC TEXT PROCESSING SYSTEM. The TSC Relocator program is required.

- 1) Load the RELOCATOR
- 2) Move it to \$3800 and set its stack pointer to \$3FFF (location \$3801 after relocation must be changed to \$3F)
- 3) Load the TSC TEXT EDITING SYSTEM
- 4) Load the program called 'LPR' which has been included with this documentation. Type in all code generated by that program.
- 5) Relocate the Editor-LPR pair according to the instructions given below. (Begin execution of the RELOCATOR at \$3800)
- 6) Load the TSC TEXT PROCESSING SYSTEM
- 7) Change the exit address of the processor (location \$0209) to \$32E6.
- 8) Begin execution at \$2000. See the LPR program for instructions on use and on adapting to a system larger than 16K.
- 9) See next page for Relocator address information

TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3

RELOCATING THE EDITOR-LPR PAIR

BEGIN ADDRESS: 0200
END ADDRESS: 1500
MOVE TO: 2000

ALTER RANGE? Y
BEGIN ADDRESS: 01FF
END ADDRESS: 1500

DATA BLOCKS? Y
BEG ADDR: 0212
END ADDR: 0354
044C
044D
0458 1491
045E 14BC
0464 14E3
0470 14E5
0476 150B
0482 150C
0946
0955
0982
0988
0A31
0A47
0BF2
0C07
0C77
0C86
0D7F
0DCA
0FC8
0FD3
10B4
10CF
1241
125B

FIX FDB19? Y
ADDRESS 021E 02CE
021F 02D2
0228 02D9
022C 02E4
0235 02EA
023C 02F4
0241 02F8
0245 02FF
024E 0305
0252 030C
0258 0310
0261 0316
0268 031C
026C 0320
0272 0329
0278 032D
027F 0335
0283 0339
028C 033D
0292 0344
0299 0348
029E 0949
02A5 094F
02AF 0953
02B4 1245
02B8 124C
02C2 1252
02C6 1259

LPR

TSC MNEMONIC ASSEMBLER PAGE 1

* TSC TEXT EDITOR-PROCESSOR CORESIDENT LINK (LPR)
*
* COPYRIGHT (C) 1977 BY
* TECHNICAL SYSTEMS CONSULTANTS, INC
* P. O. BOX 2574; W. LAFAYETTE, IN 47906
* (317) 742-7509
*
* THE PURPOSE OF THIS PROGRAM IS TO SETUP THE NECESSARY
* POINTERS IN THE TSC TEXT PROCESSOR AND TO SAVE CERTAIN
* POINTERS OF THE TSC EDITOR TO ALLOW THEM TO RUN CO-
* RESIDENT. THE ONLY CHANGE REQUIRED IN THE PROCESSOR
* TO SET THE EXIT ADDRESS (SETUP AT \$020A) TO \$32E6,
* WHICH IS THE ADDRESS OF 'EDITOR' IN 'LPR'. WHEN IT
* THE EDITOR AND READY TO PROCESS YOUR FILE, TYPE 'LF'
* FOR 'LINK PROCESSOR1'. WHEN THE PROCESSING OF THE FILE
* IS COMPLETE, CONTROL WILL RETURN TO THE EDITOR. USE
* LPR DELETES THE LOG COMMAND FROM THE EDITOR. YOU CAN
* STILL USE 'STOP' TO EXIT. THIS PROGRAM IS SETUP FOR
* 16K SYSTEM. IF YOU HAVE A LARGER SYSTEM, CHANGE ME
* IN THE EDITOR (\$0212 BEFORE RELOCATION; \$2012 AFTER)
*

* CHANGES TO EDITOR

0212	ORG	\$0212	
0212 3F FF	FDB	\$3FFF	SET MEMORY END
028E	ORG	\$028E	
028E 4C	FCC	'LPR'	PUT LPR IN COMMAND TABLE
028F 50 52			
0291 00	FCB	0	
0292 14 BD	FDB	LPR	
0358	ORG	\$0358	
0358 CE 15 0C	LDX	#BEGPT2	SETUP NEW BEGIN PTR

* EXTERNAL EQUATES

1491	ORG	\$1491	
0200	PROCSS	EQU	\$0200
009D	JNKCNT	EQU	\$009D
08A8	TSTEMP	EQU	\$08A8
005E	ZONE1	EQU	\$005E
0060	ZONE2	EQU	\$0060
006A	NUMFLG	EQU	\$006A
008F	INZFLG	EQU	\$008F
008B	BUFFER	EQU	\$00BB
0058	SPCPCT1	EQU	\$0058
0096	HEDCNT	EQU	\$0096
0040	TEMP	EQU	\$0040
0C62	MAKSPS	EQU	\$0C62
0203	RESTRRT	EQU	\$0203

LPR

TSC MNEMONIC ASSEMBLER PAGE 2

* TEMPORARY STORAGE

1491	ZONE1X	RMB	2
1493	ZONE2X	RMB	2
1495	NMFG2	RMB	2
1497	TMPEND	RMB	37
14BC	TMPBEG	-PMB	1

* ENTRY POINT UPON EXITING THE EDITOR

14BD BD 08 A8	LPR	JSR	TSTEMP	IS FILE EMPTY?
14C0 86 03		LDA R	#3	SET LINE BYTE COUNT
14C2 97 9D		STR A	JNKCNT	
14C4 CE 00 BB		LDX	#BUFFER	SAVE EDITOR DATA
14C7 DF 58		STX	SPCPT1	
14C9 CE 00 96		LDX	#HEDCNT	
14CC DF 40		STX	TEMP	
14CE CE 14 BC		LDX	#TMPBEG	
14D1 BD 0C 62		JSR	MAKSP5	
14D4 DE 5E		LDX	ZONE1	SAVE ZONE1
14D6 FF 14 91		STX	ZONE1X	
14D9 DE 60		LDX	ZONE2	SAVE ZONE2
14DB FF 14 93		STX	ZONE2X	
14DE DE 6A		LDX	NUMFLG	SAVE NUMBER & VERIFY
14E0 FF 14 95		STX	NMFG2	
14E3 7E 02 00		JMP	PROCESS	JUMP TO THE PROCESSOR

* REENTRY POINT ON EXIT FROM ASSEMBLER

14E6 4F	EDITOR	CLR A		CLEAR FLAG
14E7 97 8F		STA A	INZFLG	
14E9 CE 14 BC		LDX	#TMPBEG	RESTORE EDITOR DATA
14EC DF 58		STX	SPCPT1	
14EE CE 14 97		LDX	#TMPEND	
14F1 DF 40		STX	TEMP	
14F3 CE 00 BB		LDX	#BUFFER	
14F6 BD 0C 62		JSR	MAKSP5	
14F9 FE 14 91		LDX	ZONE1X	RESTORE ZONE1
14FC DF 5E		STX	ZONE1	
14FE FE 14 93		LDX	ZONE2X	RESTORE ZONE2
1501 DF 60		STX	ZONE2	
1503 FE 14 95		LDX	NMFG2	RESTORE NUMBER & VERIFY
1506 DF 6A		STX	NUMFLG	
1508 7E 02 03	RSTART	JMP	RESTRT	JUMP INTO EDITOR

150B 0D FCB \$0D

150C BEGPT2 EOU * START OF FILESPACE

END

NO ERROR(S) DETECTED

LPR

TSC MNEMONIC ASSEMBLER PAGE 3

SYMBOL TRBLE:

BEGPT2 150C	BUFFER 00BB	EDITOR 14E6	HEDCNT 0096	INZFLG 008F
JNKCNT 009D	LPR 14BD	MAKSP5 0C62	NMFG2 1495	NUMFLG 006A
PROCSS 0200	RESTRRT 0203	RSTART 1508	SPCPT1 0058	TEMP 0040
TMPBEG 14BC	TMPEND 1497	TSTEMP 08A8	ZONE1 005E	ZONE1X 1491
ZONE2 0060	ZONE2X 1493			

OBJECT CODE:

S1 05 0212 3F FF A8
S1 09 028E 4C 50 52 00 14 BD A7
S1 06 0358 CE 15 0C AF
S1 13 14BD BD 08 A8 86 03 97 9D CE 00 BB DF 58 CE 00 96 DF EE
S1 13 14CD 40 CE 14 BC BD 0C 62 DE 5E FF 14 91 DE 60 FF 14 D1
S1 13 14DD 93 DE 6A FF 14 95 7E 02 00 4F 97 8F CE 14 BC DF 06
S1 13 14ED 58 CE 14 97 DF 40 CE 00 BB BD 0C 62 FE 14 91 DF C5
S1 12 14FD 5E FE 14 93 DF 60 FE 14 95 DF 6A 7E 02 03 0D 1A
S9

**TSC TEXT PROCESSOR USER'S MANUAL
VERSION 2.3**

Source Listing

```

*
*
* TSC 6800 TEXT PROCESSING SYSTEM
*
*
* COPYRIGHT 1977 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* BOX 2574
* WEST LAFAYETTE, IN 47906
*

```

0040 ORG \$0040

* TEMPORARY STORAGE

* NUMBER REGISTERS

0040	NMREGS	RMB	2	A-B
0042	COLCNT	RMB	1	C
0043	DAY	RMB	1	D
0044		RMB	2	E-F
0046	GCNT	RMB	1	G
0047		RMB	1	H
0048	IND	RMB	1	I
0049		RMB	2	J-K
004B	LLN	RMB	1	L
004C	MNTH	RMB	1	M
004D	LINCNT	RMB	1	N
004E	LFM	RMB	1	O
004F	PGL	RMB	1	P
0050		RMB	5	Q-U
0055	LDIV	RMB	1	V
0056		RMB	2	W-X
0058	YEAR	RMB	1	Y
0059		RMB	1	Z

* SINGLE STORAGE

005A	GDNUM	RMB	1
005B	ADD	RMB	1
005C	SUB	RMB	1
005D	BNUM	RMB	1
005E	NPGN	RMB	1
005F	INC	RMB	1

* MACRO SAVE BLOCK

0060	NUMPNT	RMB	2
0062	EXCHR	RMB	1
0063	LSTNUM	RMB	2

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 2

0065	CMFLG	RMB	1
0066	MBFLG	RMB	1
0067	MBFPNT	RMB	2
0069	NOCR	RMB	1
006A	DONE	RMB	1
006B	FLBF	RMB	1
006C	ATFLG	RMB	1

* MORE SINGLE STORAGE

006D	LEFT	RMB	1
006E	TFILF	RMB	1
006F	NOFL	RMB	1
0070	INNUM	RMB	1
0071	NEG	RMB	1
0072	SIGN	RMB	1
0073	NSP	RMB	1
0074	PGN	RMB	1
0075	PRSCHR	RMB	1
0076	SPSPF	RMB	1
0077	DOCAP	RMB	1
0078	DOCm	RMB	1
0079	NOOUT	RMB	1
007A	TOUTL	RMB	1
007B	PTFL	RMB	1
007C	SIN	RMB	1
007D	MINDIS	RMB	1
007E	EV	RMB	1
007F	NOEXP	RMB	1
0080	NXTTAB	RMB	2
0082	TABFLG	RMB	1
0083	COLCN2	RMB	1
0084	IND2	RMB	1
0085	NXTTRP	RMB	2
0087	SVDSPC	RMB	1
0088	FINMAC	RMB	1
0089	NEGT	RMB	1
008A	IFFLG	RMB	1
008B	MACCNT	RMB	1
008C	PASFLG	RMB	1
008D	NONUMS	RMB	1
008E	DWFLG	RMB	1
008F	DFMFLG	RMB	1
0090	SPIFLG	RMB	1
0091	DIVFLG	RMB	1
0092	DIVFL2	RMB	1
0093	PRNTR	RMB	1
0094	TLPP	RMB	1
0095	LOWPG	RMB	1
0096	HIPG	RMB	1
0097	FSTRAM	RMB	2
0099	LSTRAM	RMB	2
009B	NXTRAM	RMB	2

FILE POINTERS

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 3

009D	JNKCNT	RMB	1	JUNK COUNT
009E	SBFLG	RMB	1	
009F	LLN2	RMB	1	
00A0	MACNAM	RMB	2	
00A2	MACTMP	RMB	2	
00A4	LSTAVL	RMB	2	
00A6	FSTAVL	RMB	2	
00A8	STPOUT	RMB	2	
00AA	TCPNT	RMB	2	
00AC	NXTMAC	RMB	2	
00AE	NXTOUT	RMB	2	
00B0	XMAC	RMB	2	
00B2	TSIN	RMB	1	
00B3	TIND	RMB	1	
00B4	TLLN	RMB	1	
00B5	SUPL	RMB	1	
00B6	SWRDF	RMB	1	
00B7	CAP	RMB	1	
00B8	SCAP	RMB	1	
00B9	TPOS	RMB	1	
00BA	DELIM	RMB	1	
00BB	TCNT	RMB	1	
00BC	MCNT	RMB	1	
00BD	TTLPNT	RMB	2	
00BF	ENDLIN	RMB	1	
00C0	TAB	RMB	1	
00C1	TFILL	RMB	1	

* ENVIRONMENT PARAMETERS

00C2	AUTO	RMB	2	
00C4	ROM	RMB	2	
00C6	WIDTH	RMB	2	
00C8	FILFLG	RMB	2	
00CA	PFLG	RMB	2	
00CC	PCHAR	RMB	2	
00CE	CNJ	RMB	2	
00D0	RTJ	RMB	2	
00D2	MSC	RMB	2	
00D4	CNTFLG	RMB	2	
00D6	JUST	RMB	2	
00D8	TLN	RMB	2	
00DA	BUFPNT	RMB	4	
00DE	BUFEND	RMB	4	
00E2	EBFEND	RMB	4	
00E6	CMPNPNT	RMB	2	
00E8	SPCPNT1	RMB	2	
00EA	SPCPNT2	RMB	2	
00EC	TEMP	RMB	2	
00EE	TEMP2	RMB	2	
00F0	TEMP5	RMB	2	
00F2	TEMP6	RMB	2	

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 4

00F4	RETREG	RMB	2
00F6	INDEX	RMB	2
00F8	XTEMP	RMB	2
00FA	MACEND	RMB	2
00FC	CRF	RMB	1
0110		ORG	\$0110
0110	TABS	RMB	20
0124	TABEND	RMB	1
0125	NUM	RMB	12
0200		ORG	\$0200

*
* PROGRAM ENTRY POINT

0200.7E 02 12 START JMP INTRO

* JUMP TABLE

0203 7E E1 D1	DUTCH	JMP	\$E1D1
0206 7E E1 AC	INCH	JMP	\$E1AC
0209 7E E8 D0	MON	JMP	\$E8D0
020C 7E 15 A4	PINIT	JMP	PRNIT
020F 7E 15 C4	POUCH	JMP	PROUCH
01FF	STACK	EQU	\$01FF

* MAIN PROGRAM STARTS HERE

0212 8E 01 FF	INTRO	LDS	#STACK	*** SETUP STACK ***
0215 DE 97		LDX	FSTRAM	POINT TO FIRST OF FILE
0217 DF 9B		STX	NXTRAM	SAVE IT
0219 BD 02 A8	INTRO0	JSR	CLRSPC	GO CLEAR SPACE
021C 97 95		STA A	LOWPG	SET PAGE LIMITS
021E 97 93		STA A	PRNTP	
0220 97 94		STA A	TLPP	
0222 4A		DEC A		
0223 97 96		STA A	HIPG	
0225 CE 14 EA		LDX	#CPRSTR	OUTPUT MESSAGE
0228 BD 14 90		JSR	PSTRNG	
022B 08		INX		
022C BD 14 90		JSR	PSTRNG	
022F BD 14 85		JSR	CRLF	OUTPUT CR AND LF
0232 CE 15 1A		LDX	#DATSTR	PROMPT FOR DATE
0235 BD 14 90		JSR	PSTRNG	
0238 BD 14 44		JSR	GIBUF	GET DATE
023B 7C 00 65		INC	CMFLG	
023E BD 12 1A		JSR	CHKNUM	CHECK IF VALID
0241 24 16		BCC	INTRO3	
0243 96 70		LDA A	INNUM	
0245 97 4C		STA A	MNTH	GET MONTH & SAVE
0247 BD 12 1A		JSR	CHKNUM	CHECK FOR DAY
024A 24 0D		BCC	INTRO3	

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 5

024C 96 78	LDA A	INNUM	GET & SAVE	
024E 97 43	STR A	DAY		
0250 BD 12 1A	JSR	CHKNUM	CHECK FOR YEAR	
0253 24 04	BCC	INTRO3		
0255 96 70	LDA A	INNUM	GET & SAVE	
0257 97 58	STA A	YEAR		
0259 CE 15 2C	INTRO3	LDX	#PRQU	PROMPT FOR PRINTER
025C BD 14 90	JSR	PSTRNG		
025F BD 02 06	JSR	INCH	GET RESPONSE	
0262 84 50	CMP R	#'P		
0264 26 04	BNE	INTRO4		
0266 97 93	STA A	PRNTR	SET PRINTER FLAG	
0268 20 15	BRA	INTROS		
026A CE 15 66	INTRO4	LDX	#LPPSTR	LINES PER SCREEN?
026D BD 14 90	JSR	PSTRNG		
0270 BD 14 44	JSR	GIBUF	GET RESPONSE	
0273 7C 00 65	INC	CMFLG		
0276 BD 12 1A	JSR	CHKNUM	CHECK IF NUMBER	
0279 24 04	BCC	INTROS		
027B 96 70	LDA A	INNUM	GET AND SAVE	
027D 97 94	STA A	TLPP		
027F CE 15 4A	INTROS	LDX	#PGSTR	PROMPT FOR PAGES
0282 BD 14 90	JSR	PSTRNG		
0285 BD 14 44	JSR	GIBUF	GET RESPONSE	
0288 7C 00 65	INC	CMFLG		
028B BD 12 1A	JSR	CHKNUM	CHECK NUMBER	
028E 24 0D	BCC	INTRO6		
0290 96 70	LDA A	INNUM	GET AND SAVE	
0292 97 95	STA A	LOWPG		
0294 BD 12 1A	JSR	CHKNUM	CHECK HIGH PAGE	
0297 24 04	BCC	INTRO6		
0299 96 70	LDA A	INNUM	GET IT	
029B 97 96	STA A	HIPG		
029D BD 14 85	INTRO6	JSR	CRLF	OUT CR & LF
02A0 4F	CLR A			
02A1 CE 00 5A	LDX	#GDNUM	CLEAR SPACE	
02A4 BD 06	BSR	CLRSP2		
02A6 20 18	BRA	INIT	GO INITIALIZE	

* CLEAR TEMPORARY SPACE

02A8 4F	CLRSPC	CLR A		
02A9 CE 00 40	LDX	#NMREGS	SET POINTER	
02AC A7 00	CLRSP2	STA A	0,X	CLEAR SPACE
02AE 08	INX			BUMP POINTER
02AF 80 00 93	CPX	#PRNTR		FINISHED?
02B2 26 F8	BNE	CLRSP2		
02B4 CE 00 9E	LDX	#SBFLG		DO SECOND BLOCK
02B7 A7 00	CLRSP4	STA A	0,X	
02B9 08	INX			
02BA 8C 00 DA	CPX	#BUFPNT		
02BD 26 F8	BNE	CLRSP4		
02BF 39	RTS		RETURN	

* INITIALIZATION AND SETUP

02C0 CE 01 10	INIT	LDX	#TABS	SET POINTER
02C3 4F		CLR A		
02C4 R7 00	INIT25	STA A	0, X	CLEAR TABS
02C6 08		INX		
02C7 8C 01 25		CPX	#NUM	
02CA 26 F8		BNE	INIT25	FINISHED?
02CC 4C		INC A		
02CD 97 D6		STA A	JUST	SET INITIAL PARAMS.
02CF 97 D7		STA A	JUST+1	
02D1 97 BF		STA A	ENDLIN	MARK END LINE
02D3 97 FC		STA A	CRF	
02D5 97 C8		STA A	FILFLG	SET FOR FILL
02D7 97 C9		STA A	FILFLG+1	
02D9 97 4D		STA A	LINCNT	INIT LINE COUNT
02DB 97 42		STA A	COLCNT	
02DD 97 83		STA A	COLCN2	SET COLUMN CNT
02DF 97 74		STA A	PGN	SET PAGE
02E1 86 41		LDA A	#65	
02E3 97 C6		STA A	WIDTH	SET PAGE WIDTH
02E5 97 C7		STA A	WIDTH+1	
02E7 97 4B		STA A	LLN	AND LINE LENGTH
02E9 97 9F		STA A	LLN2	
02EB 97 D8		STA A	TLN	SET TITLE LENGTH
02ED 97 D9		STA A	TLN+1	
02EF 4C		INC A		
02F0 97 4F		STA A	PGL	SET PAGE LENGTH
02F2 CE 19 F4		LDX	#MACROS	
02F5 DF AC		STX	NXTMAC	INIT MACRO SPACE
02F7 DF A6		STX	FSTAVL	
02F9 86 FF		LDR A	#\$FF	
02FB R7 00	INIT3	STA A	0, X	
02FD 08		INX		
02FE 8C 1F F2		CPX	#LMACRO	FINISHED?
0301 26 F8		BNE	INIT3	
0303 DF A4		STX	LSTAVL	
0305 09		DEX		
0306 6F 00		CLR	0, X	SET END OF MACROS
0308 6F 01		CLR	1, X	
030A 6F 02		CLR	2, X	
030C 86 0D		LDA A	#\$D	FIX BUFFER
030E B7 18 0A		STA A	CMNDBF	
0311 86 A8		LDA A	#\$A8	SET FILL CHAR.
0313 97 C1		STA A	TFILL	
0315 CE 17 D8		LDX	#TRAPS	
0318 86 FF		LDA A	#\$FF	INIT TRAPS
031A R7 00	INIT4	STA A	0, X	
031C 08		INX		
031D 8C 18 0B		CPX	#TRPEND	FINISHED?
0320 26 F8		BNE	INIT4	
0322 BD 02 0C		JSR	PINIT	INIT PRINTER
0325 CE 15 E4		LDX	#LINBUF	

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 7

0328 DF DA	STX	BUFPNT	SET POINTER
032A DF DC	STX	BUFPNT+2	
032C BD 06 46	JSR	FIXBFE	FIX BUFFER END
032F DE DE	LDX	BUFEND	
0331 DF E0	STX	BUFEND+2	
0333 CE 16 7F	LDX	#EXTBUF	
0336 DF E4	STX	EBFEND+2	
0338 CE 19 72	LDX	#MACTBL	CLEAR MACRO TABLE
033B DF FA	STX	MACEND	

* MAIN PROCESSOR LOOP

033D 96 74	PROC	LDA A	PGN	CHECK PAGE NUMBER
033F 91 95		CMP A	LOWPG	AGAINST LOW PAGE
0341 24 06	<i>jmp</i>	(SBS)	PROC3	8cc
0343 C6 0F	PROC2	LDA B	#\$F	
0345 D7 79		STA B	NOOUT	SET NO OUTPUT FLAG
0347 20 08		BRA	PUNTST	
0349 91 96	PROC3	CMP A	HIPG	AGAINST HIGH PAGE
034B 23 03		BLS	PROC4	
034D 7E 09 53		JMP	FINIS4	IF PAST, FINISH
0350 7F 00 79	PROC4	CLR	NOOUT	

* TEST FOR PUNCTUATION

0353 96 CR	PUNTST	LDA A	PFLG	TEST FLAG
0355 81 03		CMP A	#3	
0357 26 07		BNE	PUNTS3	
0359 96 CC		LDA A	PCHAR	GET SPARE CHAR.
035B 7F 00 CR	PUNTS2	CLR	PFLG	CLEAR PUNCT. FLAG
035E 20 37		BRA	JSTFY	
0360 BD 06 8C	PUNTS3	JSR	GETCHR	GET NEXT CHAR.
0363 D6 6A		LDA B	DONE	FINISHED?
0365 27 03		BEQ	PUNT35	
0367 7E 09 51		JMP	FINISH	
0368 D6 C8	PUNT35	LDA B	FILFLG	FILL ON?
036C 27 29		BEQ	JSTFY	
036E D6 CR		LDA B	PFLG	TEST PUNCT. FLAG
0370 C1 01		CMP B	#1	
0372 22 19		BHI	PUNTS7	
0374 27 11		BEQ	PUNTS6	
0376 81 2E		CMP A	#`	IS CHAR.A ` . ` ?
0378 27 08		BEQ	PUNTS4	
037A 81 21		CMP A	#`!	IS IT ` !` ?
037C 27 04		BEQ	PUNTS4	
037E 81 3F		CMP A	#`?	IS IT ` ?` ?
0380 26 03		BNE	PUNTS5	
0382 7C 00 CR	PUNTS4	INC	PFLG	SET PUNCT. FLAG
0385 20 10	PUNTS5	BRA	JSTFY	
0387 81 20	PUNTS6	CMP A	#\$20	IS CHAR SPACE?
0389 27 F7		BEQ	PUNTS4	
038B 20 CE		BRA	PUNTS2	
038D 81 20	PUNTS7	CMP A	#\$20	CHECK FOR SPACE
038F 27 CR		BEQ	PUNTS2	

0391 97 CC	STA A	PCHAR	SAVE SPARE CHAR.
0393 86 20	LDA A	#\$20	SET FOR SPACE
0395 20 EB	BRA	PUNTS4	

* JUSTIFICATION LOOP

0397 CE 16 7F	JSTFY	-LDX	#EXTBUF	FIX EXTRA POINTERS
0398 DF EE		STX	TEMP2	
039C DF E2		STX	EBFPND	
039E DE DA		LDX	BUFPNT	GET BUFFER POINTER
03A0 81 0D		CMP A	#\$D	IS CHAR. A CR?
03A2 26 14		BNE	JSTFY3	
03A4 D6 C8		LDA B	FILFLG	FILL MODE?
03A6 26 05		BNE	JSTFY2	
03A8 DF A8	JSTFY1	STX	STPOUT	MARK LAST BUF. POS.
03AA 7E 05 BB		JMP	OUTLIN	OUTPUT LINE
03AD 86 20	JSTFY2	LDA A	#\$20	GET A SPACE
03AF A7 00	JSTF25	STA A	0,X	SAVE IT
03B1 08		INX	BUFEND	BUMP POINTER
03B2 9C DE		CPX	BUFEND	END OF BUFFER?
03B4 26 F9		BNE	JSTF25	
03B6 20 1E		BRA	JSTFY6	
03B8 A7 00	JSTFY3	STA A	0,X	SAVE CHARACTER
03BA 7C 00 42		INC	COLCNT	BUMP COLUMN COUNT
03BD 08		INX		BUMP POINTER
03BE 9C DE		CPX	BUFEND	END?
03C0 26 06		BNE	JSTFY4	
03C2 D6 C8		LDA B	FILFLG	FILL MODE?
03C4 27 02		BEQ	JSTFY4	
03C6 20 0E		BRA	JSTFY6	
03C8 8C 16 7F	JSTFY4	CPX	#EXTBUF	BUFFER OVERFLOW?
03CB 26 04		BNE	JSTFY5	
03CD 86 0D		LDA A	#\$D	STUFF A C.R.
03CF 20 D7		BRA	JSTFY1	
03D1 DF DA	JSTFY5	STX	BUFPNT	SAVE BUF POINTER
03D3 7E 03 3D	JSTF55	JMP	PROC	REPEAT LOOP
03D6 D6 CA	JSTFY6	LDA B	PFLG	CHECK FLAG
03D8 C1 03		CMP B	#3	
03DA 26 04		BNE	JSTF63	
03DC 96 CC		LDA A	PCHAR	GET CHARACTER
03DE 20 0B		BRA	JSTF65	
03E0 81 20	JSTF63	CMP A	#\$20	IS CHAR = SPACE?
03E2 27 4E		BEQ	ADJSPC	
03E4 BD 06 8C		JSR	GETCHR	GET NEXT CHARACTER
03E7 81 20		CMP A	#\$20	IS IT SPACE?
03E9 27 47		BEQ	ADJSPC	
03EB 36	JSTF65	PSH A		SAVE CHAR.
03EC 86 20		LDA A	#\$20	
03EE DE DE		LDX	BUFEND	GET TO END
03F0 09	JSTFY7	DEX		
03F1 8C 15 E3		CPX	#LINBUF-1	LOOK FOR SPACES
03F4 27 1C		BEQ	JSTFY9	
03F6 A1 00		CMP A	0,X	
03F8 26 F6		BNE	JSTFY7	

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 9

03FA 08	JSTFY8	INX	BUMP POINTER	
03FB 9C DE		CPX	BUFEND	
03FD 27 13		BEQ	JSTFY9	
03FF A6 00		LDA R	0,X	PICK UP CHARACTER
0401 DF EC		STX	TEMP	SAVE X
0403 DE EE		LDX	TEMP2	
0405 A7 00		STA R	0,X	MOVE THE CHAR.
0407 08		INX		
0408 DF EE		STX	TEMP2	
040A DE EC		LDX	TEMP	RESTORE X
040C 86 20		LDA R	#\$20	SET WITH SPACE
040E A7 00		STA R	0,X	SAVE IT
0410 20 E8		BRA	JSTFY8	REPEAT
0412 32	JSTFY9	PUL A		RESTORE CHARACTER
0413 7F 00 82		CLR	TABFLG	CLEAR TABS
0416 CE 01 24		LDX	#TABEND	POINT TO TABS
0419 DF 80		STX	NXTTAB	SET NEXT TAB
041B DE EE		LDX	TEMP2	RESTORE X
041D A7 00	JSTF95	STA R	0,X	SAVE LAST CHAR.
041F 08		INX		BUMP POINTER
0420 DF E2		STX	EBFEND	SET END
0422 81 20		CMP R	#\$20	WAS CHAR A SPACE?
0424 27 0C		BEQ	ADJSPC	
0426 8C 16 AC		CPX	#LINBU2	BUFFER OVERFLOW?
0429 27 07		BEQ	ADJSPC	
042B 3D 26 8C		JSR	GETCHR	GET NEXT CHAR.
042E DE E2		LDX	EBFEND	GET POINTER
0430 29 EB		BRA	JSTF95	

* ADJUST BUFFER FOR SPACES

0432 5F	ADJSPC	CLR B	CLEAR COUNT
0433 CE 15 E4		LDX #LINBUF	POINT TO BUF BEGIN
0436 DF E8		STX SPCPT1	
0438 A6 00	ADJSP2	LDA R 0,X	LOOK FOR SPACES
043A 81 20		CMP R #\$20	
043C 26 09		BNE ADJS35	
043E 5C		INC B	INC THE COUNTER
043F 08		INX	BUMP POINTER
0440 9C DE		CPX BUFEND	
0442 26 F4		BNE ADJSP2	
0444 7E 05 BB	ADJSP3	JMP OUTLIN	OUTPUT LINE
0447 DF EA	ADJS35	STX SPCPT2	SET END
0449 BD 05 B7		JSR DELCHR	DELETE INIT. SPACES
044C CE 15 E4		LDX #LINBUF	POINT TO BEGIN
044F 86 20		LDA R #\$20	CHECK MORE SPACES
0451 A1 00	ADJSP4	CMP R 0,X	
0453 27 07		BEQ ADJSP5	
0455 08		INX	BUMP TIL FIND
0456 9C DE		CPX BUFEND	END OF BUFFER?
0458 27 10		BEQ ADJSP6	
045A 20 F5		BRA ADJSP4	REPEAT
045C 08	ADJSP5	INX	BUMP POINTER
045D 9C DE		CPX BUFEND	FINISHED?

045F 26 05		BNE	ADJS55	
0461 7C 00 B6		INC	SWRDF	SET SINGLE WORD
0464 20 04		BRA	ADJSP6	
0466 A1 00	ADJS55	CMP A	0, X	CHECK NEXT CHAR.
0468 27 F2		BEQ	ADJSP5	
046A D6 D4	ADJSP6	LDA B	CNTFLG	CENTERING?
046C 27 03		BEQ	ADJSP7	
046E 7E 06 19		JMP	CNTRIT	GO CENTER LINE
0471 D6 D6	ADJSP7	LDA B	JUST	JUSTIFICATION?
0473 27 DF		BEQ	ADJSP3	
0475 D6 D0		LDA B	RTJ	RIGHT HAND?
0477 27 03		BEQ	ADJSP8	
0479 7E 05 A7		JMP	RIGHTJ	GO DO RIGHT
047C D6 CE	ADJSP8	LDA B	CNJ	CENTER JUST. ?
047E 27 03		BEQ	ADJSP9	
0480 7E 05 B2		JMP	CENTJ	GO CENTER
0483 D6 B6	— ADJSP9	LDA B	SWRDF	CHECK SINGLE
0485 26 BD		BNE	ADJSP3	
0487 D6 6B		LDA B	FLBF	FLUSHING BUFFER?
0489 26 B9		BNE	ADJSP3	
048B D6 6D		LDA B	LEFT	WHICH SIDE
048D 27 3A		BEQ	RINS	GO FROM RIGHT

-* INSERT SPACES FROM LEFT

048F CE 15 E4	LINS	LDX	#LINSUF	SET POINTER
0492 DF EC		STX	TEMP	SAVE IT
0494 DE DE	LINS2	LDX	BUFEND	POINT TO END
0496 09		DEX		DEC THE POINTER
0497 A6 00		LDA R	0, X	GET CHARACTER
0499 81 20		CMP A	#\$20	IS IT A SPACE?
049B 26 A7		BNE	ADJSP3	
049D DE EC		LDX	TEMP	RESTORE POINTER
049F A6 00	LINS3	LDA R	0, X	GET CHAR
04A1 81 20		CMP A	#\$20	IS IT SPACE?
04A3 27 07		BEQ	LINS4	
04A5 08		INX		BUMP POINTER
04A6 9C DE		CPX	BUFEND	END OF BUFFER
04A8 27 E5		BEQ	LINS	
04AA 20 F3		BRA	LINS3	REPEAT
04AC C6 01	LINS4	LDA B	#1	SET COUNT = 1
04AE BD 05 DD		JSR	INSSPC	GO INSERT SPACE
04B1 D6 CE		LDA B	CNJ	CENTER JUST?
04B3 27 01		BEQ	LINS5	
04B5 39		RTS		RETURN
04B6 DE EC	LINS5	LDX	TEMP	RESTORE POINTER
04B8 A6 00	LINS6	LDA A	0, X	GET CHARACTER
04BA 81 20		CMP A	#\$20	IS IT SPACE?
04BC 26 07		BNE	LINS7	
04BE 08		INX		BUMP POINTER
04BF 9C DE		CPX	BUFEND	END OF BUFFER?
04C1 27 CC		BEQ	LINS	
04C3 20 F3		BRA	LINS6	
04C5 DF EC	LINS7	STX	TEMP	SAVE X

04C7 20 CB

SPA LINS2 REPEAT

* INSERT SPACES FROM RIGHT SIDE

04C9 DE DE	RINS	LDX	BUFEND	SET POINTER
04CB 86 20		LDA A	#\$20	SET UP SPACE
04CD 09	RINS2	DEX		
04CE A1 30		CMP A	0,X	IS CHAR A SPACE?
04DD 27 FB		BEQ	RINS2	
04D2 DF EC		STX	TEMP	SAVE POINTER
04D4 DE DE	RINS3	LDX	BUFEND	GO TO END
04D6 09		DEX		
04D7 A6 00		LDA A	0,X	GET CHAR.
04D9 81 20		CMP A	#\$20	IS IT SPACE?
04DB 26 2E		BNE	OUTLIN	
04DD DE EC		LDX	TEMP	RESTORE X
04DF A6 00	RINS4	LDA A	0,X	GET CHAR
04E1 81 20		CMP A	#\$20	IS IT SPACE?
04E3 27 08		BEQ	RINS5	
04E5 09		DEX		DEC THE POINTER
04E6 8C 15 E3		CPX	#LINBUF-1	FINISHED?
04E9 27 DE		BEQ	RINS	
04EB 20 F2		BRA	RINS4	REPEAT
04ED C6 01	RINS5	LDA B	#1	SET COUNT = 1
04EF BD 05 DD		JSR	INSSPC	INSERT SPACE
04F2 D6 CE		LDA B	CNJ	CENTER JUST?
04F4 27 01		BEQ	RINS6	
04F6 39		RTS		RETURN
04F7 DE EC	RINS6	LDX	TEMP	RESTORE POINTER
04F9 A6 00	RINS7	LDA A	0,X	GET CHARACTER
04FB 81 20		CMP A	#\$20	SPACE?
04FD 26 08		BNE	RINS8	
04FF 09		DEX		
0500 8C 15 E3		CPX	#LINBUF-1	FINISHED?
0503 27 C4		BEQ	RINS	
0505 20 F2		BRA	RINS7	REPEAT
0507 DF EC	RINS8	STX	TEMP	SAVE POINTER
0509 20 C9		BRA	RINS3	

* OUTPUT LINE FROM WORK BUFFER

0508 7F 00 B6	OUTLIN	CLR	SWRDF	CLR FLAG
050E D6 4E		LDA B	LFM	LEFT MARGIN?
0510 7D 00 7B		TST	PTFL	PUT IN INDENT?
0513 26 02		BNE	OUTLI1	
0515 DB 48		ADD B	IND	ADJUST LEFT
0517 7F 00 7B	OUTLI1	CLR	PTFL	
051A DB 7C		ADD B	SIN	ADD IN SINGLE IN.
051C 28 0C		BMI	OUTLI3	
051E 27 0A		BEQ	OUTLI3	
0520 86 20	OUTLI2	LDA A	#\$20	SET UP SPACE
0522 37		PSH B		
0523 BD 14 9F		JSR	OUTCHR	OUTPUT SPACE
0526 33		PUL B		

			DEC COUNT
0527 5A		DEC B	
0528 26 F6		BNE OUTL12	
052A D6 C8	OUTL13	LDA B FILFLG	FILL MODE?
052C 27 11		BEQ OUTL15	
052E 86 20		LDA R #\$20	SETUP SPACE
0530 DE DE		LDX BUFEND	GO TO END
0532 8C 15 E4	OUTL14	CPX #LINBUF	EMPTY?
0535 27 19		BEQ OUTL16	
0537 09		DEX	DEC THE POINTER
0538 A1 00		CMP A 0, X	IS IT SPACE?
053A 27 F6		BEQ OUTL14	
053C 08		INX	BUMP POINTER
053D DF A8		STX STPOUT	SET END
053F CE 15 E4	OUTL15	LDX #LINBUF	
0542 9C A8		CPX STPOUT	EMPTY?
0544 27 0A		BEQ OUTL16	
0546 A6 00	OUTL55	LDA A 0, X	GET CHARACTER
0548 BD 14 9F		JSR OUTCHR	OUTPUT IT
054B 08		INX	BUMP POINTER
054C 9C A8		CPX STPOUT	FINISHED?
054E 26 F6		BNE OUTL55	
0550 5F	OUTL16	CLR B	CLEAR FLAGS
0551 D7 8E		STA B DWFLG	
0553 D7 6F		STA B NOFL	
0555 D7 CA		STA B PFLG	
0557 D7 73		STA B NSP	
0559 D7 7C		STA B SIN	
055B 73 00 6D		COM LEFT	SWITCH SP. SIDES
055E CE 15 E4		LDX #LINBUF	SET POINTER
0561 DF DA		STX BUFFPNT	
0563 CE 01 10		LDX #TABS	SET TABS
0566 DF 80		STX NXTTAB	
0568 BD 14 14		JSR FIXWD	GO FIX WIDTH
056B CE 16 7F	OUTL17	LDX #EXTBUF	
056E 9C E2	OUTL75	CPX EBFEND	CHECK FOR EXTRA?
0570 27 17		BEQ OUTL18	
0572 A6 00		LDA A 0, X	GET CHARACTER
0574 08		INX	
0575 DF EC		STX TEMP	
0577 DE DA		LDX BUFFPNT	TRANSFER IT
0579 A7 00		STA A 0, X	
057B 08		INX	BUMP POINTER
057C 9C DE		CPX BUFEND	CHECK END
057E 27 09		BEQ OUTL18	OVERFLOW!
0580 DF DA		STX BUFFPNT	SAVE IT
0582 DE EC		LDX TEMP	
0584 7C 00 42		INC COLCNT	BUMP COLUMN COUNT
0587 20 E5		BRA OUTL75	REPEAT
0589 CE 16 7F	OUTL18	LDX #EXTBUF	FIX POINTER
058C DF E2		STX EBFEND	
058E BD 08 99		JSR PCRLF	OUTPUT CR & LF
0591 96 D2		LDA A MSC	MULTIPLE SPACE?
0593 27 0A		BEQ OUTL85	
0595 4A	OUTL82	DEC A	

596 27 07		BEQ	OUTL85	
598 36		PSH A		OUTPUT EXTRA SPACE
599 BD 08 99		JSR	PCRLF	
59C 32		PUL A		
59D 20 F6		BRA	OUTL82	
59F 96 6B	OUTL85	LDA A	FLBF	FLUSHING?
5A1 27 81		BEQ	OUTLI9	
5A3 39		RTS		
5A4 7E 03 3D	OUTLI9	JMP	PROC	GO PROCESS

* RIGHT HAND JUSTIFY

5A7 BD 06 0B	RIGHTJ	JSR	CNTSPC	COUNT SPACES
5A9 CE 15 E3	RIGHT2	LDX	#LINBUF-1	
5AD 8D 2E		BSR	INSSPC	INSERT SPACES
5AF 7E 05 0B		JMP	OUTLIN	OUTPUT LINE

* CENTER JUSTIFY

15B2 BD 57	CENTJ	BSR	CNTSPC	COUNT SPACES
15B4 57		ASR B		DIVIDE BY 2
15B5 20 F3		BRA	RIGHT2	

* DELETE CHARACTERS

35B7 DE EA	DELCHR	LDX	SPCPY2	GET POINTER
35B9 9C E8		CPX	SPCPY1	EMPTY?
35BB 27 1F		BEQ	DELCH4	
35BD 9C DE		CPX	BUFEND	
35BF 27 0E		BEQ	DELCH3	
05C1 A6 00		LDA A	0,X	GET CHARACTER
05C3 08		INX		BUMP THE POINTER
05C4 DF EA		STX	SPCPY2	SAVE IT
05C6 DE E8		LDX	SPCPY1	RESTORE
05C8 A7 00		STA A	0,X	SAVE CHARACTER
05CA 08		INX		BUMP POINTER
05CB DF E8		STX	SPCPY1	
05CD 20 E8		BRA	DELCHR	REPEAT
05CF DE E8	DELCH3	LDX	SPCPY1	GET POINTER
05D1 86 20		LDA A	#\$20	SETUP SPACE
05D3 9C DE	DELCH3	CPX	BUFEND	
05D5 27 05		BEQ	DELCH4	
05D7 A7 00		STA A	0,X	PUT IN SPACE
05D9 08		INX		BUMP POINTER
05DA 20 F7		BRA	DELCH3	
05DC 39	DELCH4	RTS		

TICKET

* INSERT SPACES

05DD 5D	INSSPC	TST B		TEST COUNT
05DE 27 2A		BEQ	INSSP5	IF NONE, RETURN
05E0 37		PSH B		SAVE COUNT
05E1 DF EC		STX	TEMP	SAVE X
05E3 DE DE		LDX	BUFEND	POINT TO END

05E5 DF E8		STX	SPCPT1	SAVE
05E7 08	INSSP2	INX		
05E8 5A		DEC B		DEC THE COUNT
05E9 26 FC		BNE	INSSP2	
05EB DF EA		STX	SPCPT2	SAVE POINTER
05ED DE E8	INSSP3	LDX	SPCPT1	
05EF 9C EC		CPX	TEMP	FINISHED?
05F1 27 0E		BEQ	INSSP4	
05F3 A6 00		LDA A	0,X	GET CHARACTER
05F5 09		DEX		DEC THE POINTER
05F6 DF E8		STX	SPCPT1	SAVE IT
05F8 DE EA		LDX	SPCPT2	
05FA A7 00		STA A	0,X	PUT CHARACTER
05FC 09		DEX		
05FD DF EA		STX	SPCPT2	
05FF 20 EC		BRA	INSSP3	REPEAT
0601 33	INSSP4	PUL B		RESTORE COUNT
0602 86 20		LDA A	#\$20	SETUP SPACE
0604 08	INSS44	INX		BUMP THE POINTER
0605 A7 00		STA A	0,X	STUFF SPACE
0607 5A		DEC B		DEC THE COUNT
0608 26 FA		BNE	INSS44	
060A 39	INSSP5	RTS		RETURN

* COUNT SPACES

060B 5F	CNTSPC	CLR B		CLEAR COUNT
060C 86 20		LDA A	#\$20	SETUP SPACE
060E DE DE		LDX	BUFEND	SET POINTER
0610 09	CNTSP2	DEX		
0611 A1 00		CMP A	0,X	SPACE?
0613 26 03		BNE	CNTSP3	
0615 5C		INC B		BUMP THE COUNT
0616 20 F8		BRA	CNTSP2	
0618 39	CNTSP3	RTS		

* CENTER LINE

0619 8D F0	CNTRIT	BSR	CNTSPC	GO COUNT SPACES
061B 96 BE		LDA A	DWFLG	DOUBLE WIDTH?
061D 27 0E		BEQ	CNTRI4	
061F 96 C6		LDA A	WIDTH	GET WIDTH
0621 10		SBA		
0622 48		ASL A		FIX FOR DOUBLE
0623 91 C6		CMP A	WIDTH	
0625 22 0C		BHI	CNTRI5	
0627 16		TAB		SAVE
0628 96 C6		LDA A	WIDTH	
062A 10		SBA		SUB FROM WIDTH
062B 16		TAB		
062C 57		ASR B		DIVIDE BY TWO
062D 57	CNTRI4	ASR B		
062E CE 15 E3		LDX	#LINBUF-1	SET POINTER
0631 8D AA		BSR	INSSPC	GO INSERT SPACE

0633 7A 00 D4	CNTRIS	DEC	CNTFLG	DEC CENTER COUNT
0636 26 0B		BNE	CNTRI6	
0638 4F		CLR A		
0639 97 D4		STA A	CNTFLG	CLEAR FLAG
063B 96 6E		LDA A	TFILF	GET TEMP FILL
063D 97 C8		STA A	FILFLG	SET FILL
063F DE DE		LDX	BUFEND	SET POINTER
0641 DF A8		STX	STPOUT	SET END
0643 7E 05 0B	CNTRI6	JMP	OUTLIN	OUTPUT LINE

* FIX BUFFER END POINTER

0646 C6 05		FIXBFE	LDA B	#5	CHECK BUFFER POS.
0648 CE 02 25			LDX	#NUM+\$100	COMPARE WITH TEMP
064B BD 0B ED			JSR	SENV9	CHECK IT
064E 81 2D			CMP A	#\$2D	OVERFLOW??
0650 27 03			BEQ	FIXBF4	
0652 7E 12 5E			JMP	PRNU27	FINISH PROCESS
0655 CE 15 E4	FIXBF4		LDX	#LINBUF	SET POINTER
0658 DF DE			STX	BUFEND	
065A 96 4B			LDA A	LLN	GET LINE LENGTH
065C 90 C6			SUB A	WIDTH	CRLC. COLUMN NUM.
065E 4C			INC A		
065F 97 42			STA A	COLCNT	SAVE COUNT
0661 5F			CLR B		
0662 96 C6			LDA A	WIDTH	GET WIDTH
0664 9B DF			ADD A	BUFEND+1	ADD TO BUFEND
0666 D9 DE			ADC B	BUFEND	
0668 97 DF			STA A	BUFEND+1	SAVE RESULT
066A D7 DE			STA B	BUFEND	
066C 39			RTS		RETURN

* RETURN FROM MACRO

066D 7F 00 88	RETMAC	CLR	FINMAC	CLEAR FLAG
0670 32		PUL A		FIX STACK
0671 32		PUL A		
0672 32		PUL A		
0673 97 D4		STA A	CNTFLG	RESTORE FLAG
0675 CE 00 60		LDX	#NUMPNT	
0678 32	RETMAC2	PUL A		RESTORE VALUES
0679 A7 00		STA A	0, X	
067B 08		INX		
067C 8C 00 6D		CPX	#LEFT	FINISHED?
067F 26 F7		BNE	RETMAC2	
0681 7A 00 9B		DEC	MACCNT	DEC MACRO COUNTER
0684 96 6C		LDA A	ATFLG	DOING AT?
0686 27 0B		BEQ	GETCH1	
0688 39		RTS		RETURN

* CLEAR 'ENDLIN' AND GET CHARACTER

0689 7F 00 BF	CLRGET	CLR	ENDLIN	
---------------	--------	-----	--------	--

* GET NEXT CHARACTER

068C BD 14 71	GETCHR	JSR	TSTBRK	TEST FOR BREAK
068F 96 88		LDA A	FINMAC	FINISH MACRO?
0691 26 DA		BNE	RETMAC	
0693 96 62	GETCH1	LDA A	EXCHR	GET EXTRA CHAR.
0695 27 03		.BEQ	GETCH2	
0697 7E 11 B6		JMP	FTCHNM	
069A 96 65	GETCH2	LDA A	CMFLG	GET NUMBER
069C 27 0D		.BEQ	GETCH3	COMMAND?
069E DE E6	GETC22	LDX	CMNPNT	SET POINTER
06A0 A6 00		LDA A	0, X	GET CHARACTER
06A2 81 0D		CMP A	#\$D	C. R. ?
06A4 27 01		.BEQ	GETC25	
06A6 08		INX		BUMP THE POINTER
06A7 DF E6	GETC25	STX	CMNPNT	SAVE IT
06A9 20 29		BRA	FETCHR	
06AB 96 9E	GETCH3	LDA A	SBFLG	SPECIAL BUFFER?
06AD 26 EF		BNE	GETC22	
06AF 96 66		LDA A	MBFLG	MACRO BUFFER?
06B1 27 08		.BEQ	GETCH4	
06B3 BD 0E E3		JSR	INMAC	GET CHARACTER
06B6 26 1C		BNE	FETCHR	
06B8 7E 0F 7D		JMP	MCEND	FINISH MACRO
06BB 96 90	GETCH4	LDA A	SPIFLG	SPECIAL INPUT?
06BD 27 05		.BEQ	GETCH5	
06BF BD 02 06		JSR	INCH	GET CHARACTER
06C2 20 10		BRA	FETCHR	
06C4 96 8F	GETCH5	LDA A	DFMFLG	DEFINE MACRO?
06C6 9A 69		ORA A	NOCR	
06C8 26 07		BNE	GETCH6	
06CA 96 82		LDA A	TABFLG	TABS?
06CC 27 03		.BEQ	GETCH6	
06CE 7E 0B 10		JMP	DOTAB	GO DO TAB
06D1 BD 14 BA	GETCH6	JSR	INCHR	GET CHARACTER

* FETCH AND CHECK CHARACTER

06D4 81 1A	FETCHR	CMP A	#\$1A	END OF FILE?
06D6 26 05		BNE	FETCH2	
06D8 97 6A		STA A	DONE	SET FLAG
06DA 7E 09 51		JMP	FINISH	
06DD 81 0D	FETCH2	CMP A	#\$D	C. R. ?
06DF 26 29		BNE	FETCH3	
06E1 7F 00 76		CLR	SPSPF	SPECIAL SPACE?
06E4 D6 69		LDA B	NOCR	
06E6 26 32		BNE	FETC35	
06E8 D6 9E		LDA B	SBFLG	CHECK FLAG
06EA 26 07		BNE	FETC22	
06EC D6 BF		LDA B	ENDLIN	END OF LINE?
06EE 27 03		.BEQ	FETC22	
06F0 BD 09 2A		JSR	FLUSHB	FLUSH BUFFER
06F3 97 BF	FETC22	STA A	ENDLIN	SET FLAGS
06F5 7F 00 9E		CLR	SBFLG	

06F8 D6 8F		LDA B	DFMFLG	TEST
06FA DA D4		ORA B	CNTFLG	
06FC 26 08		BNE	FETC25	
06FE D6 C8		LDA B	FILFLG	TEST FILL
0700 27 04		BEQ	FETC25	
0702 86 20		LDA A	#\$20	SETUP SPACE
0704 20 19		BRA	FETC36	
0706 86 0D	FETC25	LDA A	#\$D	SETUP C. R.
0708 20 15		BRA	FETC36	
070A D6 8F	FETCH3	LDA B	DFMFLG	GET FLAG
070C DA 8C		ORA B	PASFLG	
070E 26 0A		BNE	FETC35	
0710 D6 75		LDA B	PASCHR	PASS CHAR?
0712 27 0F		BEQ	FETCH4	
0714 81 20		CMP A	#\$20	IS IT A SPACE?
0716 26 02		BNE	FETC35	
0718 8A 80		ORA A	#\$80	SET PARITY
071A .5F	FETC35	CLR B		CLEAR FLAGS
071B D7 BF		STA B	ENDLIN	
071D D7 75		STA B	PASCHR	
071F 7F 00 B7	FETC36	CLR	CAP	
0722 39	FETC37	RTS		RETURN
0723 81 1F	FETCH4	CMP A	#\$1F	CHECK CHAR
0725 22 03		BHI	FEIC45	
0727 7E 06 8C		JMP	GETCHR	GO GET CHAR.
072A D6 BF	FETC45	LDA B	ENDLIN	END OF LINE?
072C 27 1A		BEQ	FETCH5	
072E 81 2E		CMP A	#`.	PERIOD?
0730 27 06		BEQ	FETC47	
0732 81 3A		CMP A	#`:	COLON?
0734 26 05		BNE	FETC48	
0736 97 6F		STA A	NOFL	SET NO FLUSH
0738 7E 07 DF	FETC47	JMP	COMMAND	DO COMMAND
073B 81 20	FETC48	CMP A	#\$20	SPACE?
073D 26 09		BNE	FETCH5	
073F 97 76		STA A	SPSPF	SET FLAG
0741 BD 09 2A		JSR	FLUSHB	FLUSH BUFFER
0744 86 A0	FETC49	LDA A	#\$A0	
0746 20 D2		BRA	FETC35	
0748 D6 76	FETCH5	LDA B	SPSPF	TEST FLAG
074B 27 07		BEQ	FETC55	
074C 81 20		CMP A	#\$20	IS IT SPACE?
074E 27 F4		BEQ	FETC49	
0750 7F 00 76		CLR	SPSPF	CLEAR OUT
0753 D6 65	FETC55	LDA B	CMFLG	COMMAND?
0755 DA 66		ORA B	MBFLG	
0757 DA 8D		ORA B	NONUMS	
0759 DA 90		ORA B	SPIFLG	
075B DA 9E		ORA B	SBFLG	
075D 26 1C		BNE	FETCH6	
075F 91 C0		CMP A	TAB	CHECK IF TAB
0761 26 18		BNE	FETCH6	
0763 DE 80		LDX	NXTTAB	GET NEXT TAB
0765 D6 42		LDA B	COLCNT	GET COUNT

0767 6D 00	FETC57	TST	0,X	CHECK
0769 27 AF		BEQ	FETC35	
076B E1 00		CMP B	0,X	FINISHED?
076D 25 03		BLO	FETC58	
076F 08		INX		BUMP THE POINTER
0770 20 F5		BRA	FETC57	
0772 DF 80	FETC58	STX	NXTTAB	SAVE POINTER
0774 96 C1		LDA A	TFILL	
0776 97 82		STA A	TABFLG	SET FLAG
0778 7E 07 1A	FETC59	JMP	FETC35	
077B D6 8D	FETCH6	LDA B	NONUMS	NUMBERS?
077D 26 1D		BNE	FETCH7	
077F 81 23		CMP A	#'#	POUND SIGN?
0781 27 04		BEQ	FETC65	
0783 81 25		CMP A	#'%	PERCENT SIGN?
0785 26 15		BNE	FETCH7	
0787 D6 69	FETC65	LDA B	NOCR	DO C.R.?
0789 37		PSH B		
078A 97 69		STA A	NOCR	SAVE VALUES
078C 97 7F		STA A	NOEXP	
078E BD 11 9E		JSR	CLRNUM	CLEAR NUMBER
0791 BD 12 88		JSR	PRNU32	PROCESS NUMBER
0794 33		PUL B		
0795 D7 69		STA B	NOCR	RESTORE VALUES
0797 24 DF		BCC	FETC59	
0799 7E 06 8C		JMP	GETCHR	GET CHARACTER
079C 81 5C	FETCH7	CMP A	#'\`	BACK SLASH?
079E 26 05		BNE	FETC75	
07A0 97 75		STA A	PASCHR	SET PASS CHAR.
07A2 7E 06 89		JMP	CLRGET	GO GET IT
07A5 81 40	FETC75	CMP A	#'@	AT SIGN?
07A7 27 1C		BEQ	CAPIT	
07A9 81 5E		CMP A	#\$5E	UP ARROW?
07AB 27 21		BEQ	SETCAP	
07AD D6 B7		LDA B	CAP	CHECK MODE
07AF DA B8		ORA B	SCAP	
07B1 DA 66		ORA B	MBFLG	
07B3 26 C3		BNE	FETC59	
07B5 81 41		CMP A	#'A	CHECK IF LETTER
07B7 25 BF		BLO	FETC59	
07B9 81 5A		CMP A	#'Z	
07BB 22 BB		BHI	FETC59	
07BD D6 77		LDA B	DOCAP	DO CAP?
07BF 27 B7		BEQ	FETC59	
07C1 88 20		ADD A	#\$20	FORCE TO LOWER
07C3 20 B3	FETCH8	BRA	FETC59	
* CAP SINGLE LETTER				
07C5 D6 77	CAPIT	LDA B	DOCAP	CHECK MODE
07C7 27 FA		BEQ	FETCH8	
07C9 97 B7		STA A	CAP	SET FLAG
07CB 7E 06 89	CAPIT2	JMP	CLRGET	

* CAP STRING OF LETTERS

-07CE D6 7?	SETCAP	LDA B	DOCAP	CHECK MODE
-07D0 27 F1		BEQ	FETCH8	
-07D2 D6 B8		LDA B	SCAP	GET FLAG
-07D4 27 05		BEQ	SETCA2	
-07D6 7F 00 B8		CLR	SCAP	CLEAR IT
-07D9 20 F0		BRA	CAPIT2	
-07DB 97 B8	SETCA2	STA A	SCAP	SET FOR STRING
-07DD 20 EC		BRA	CAPIT2	

* COMMAND PROCESSOR

-07DF 7F 00 BF	COMAND	CLR	ENDLIN	CLEAR FLAG
-07E2 CE 18 09		LDX	#CMNDBF-1	SET POINTER
-07E5 08	COMAN2	INX		BUMP IT
-07E6 7C 00 69		INC	NOCR	SET NO C. R.
-07E9 DF F2		STX	TEMP6	SAVE POINTER
-07EB 7C 00 8D		INC	NONUMS	
-07EE BD 06 8C		JSR	GETCHR	GET CHARACTER
-07F1 DE F2		LDX	TEMP6	RESTORE POINTER
-07F3 7F 00 69		CLR	NOCR	CLEAR FLAG
-07F6 7F 00 8D		CLR	NONUMS	
-07F9 R7 00		STA A	0, X	PUT CHARACTER
-07FB 81 0D		CMP A	#\$D	WAS IT A C. R. ?
-07FD 26 E6		BNE	COMAN2	
-07FF 7F 00 BF		CLR	ENDLIN	RESET END LINE
-0802 CE 18 0A		LDX	#CMNDBF	SET POINTER
-0805 A6 00	COMAN3	LDA A	0, X	GET CHARACTER
-0807 08		INX		BUMP THE POINTER
-0808 E6 00		LDA B	0, X	GET NEXT CHAR
-080A 08		INX		BUMP
-080B DF E6		STX	CMNPNT	SAVE THE POINTER
-080D 81 5F		CMP A	#\$5F	LOWER CASE?
-080F 23 04		BLS	COMAN4	
-0811 80 20		SUB R	#\$20	SET TO UPPER
-0813 C0 20		SUB B	#\$20	
-0815 CE 09 5C	COMAN4	LDX	#CMNDT	POINT TO TABLE
-0818 A1 00	COMAN5	CMP A	0, X	COMPARE FIRST
-081A 26 0C		BNE	COMAN7	
-081C E1 01		CMP B	1, X	COMPARE SECOND
-081E 26 08		BNE	COMAN7	
-0820 97 65		STA A	CMFLG	FOUND COMMAND
-0822 EE 02		LDX	2, X	GET ADDRESS
-0824 RD 00	COMAN6	JSR	0, X	GO DO IT
-0826 20 2F		BRA	FINCM	FINISH COMMAND
-0828 08	COMAN7	INX		BUMP POINTER
-0829 08		INX		
-082A 08		INX		
-082B 08		INX		
-082C 8C 0A 3C		CPX	#TBLEND	TABLE END?
-082F 26 E7		BNE	COMAN5	
-0831 36		PSH A		
-0832 96 BB		LDA A	MACCNT	TEST MACRO NUMBER

0834 81 07		CMP A #7	
0836 32		PUL A	
0837 24 15		BHS MACOVF	OVERFLOW?
0839 CE 19 72	COMAN8	LDX #MACTBL	POINT TO MACROS
083C 9C FA		CPX MACEND	END?
083E 27 17		BEQ FINCM	
0840 A1 00		CMP A 0,X	COMPARE FIRST
0842 26 04		BNE COMAN9	
0844 E1 01		CMP B 1,X	COMPARE SECOND
0846 27 29		BEQ CALMAC	
0848 08	COMAN9	INX	FIND NEXT ENTRY
0849 08		INX	
084A 08		INX	
084B 08		INX	
084C 20 EE		BRA COMAN8	

* MACRO OVERFLOW ERROR

084E CE 15 79	MACOVF	LDX #OVFSTR	POINT TO STRING
0851 BD 14 90		JSR PSTRNG	OUTPUT IT
0854 7E 02 09		JMP MON	

* FINISH COMMAND

0857 96 8A	FINCM	LDA A IFFLG	CHECK FOR IF
0859 27 0A		BEQ FINCM1	
085B 4F		CLR A	CLEAR FLAGS
085C 97 BF		STA A ENDLIN	
085E 97 65		STA A CMFLG	
0860 97 8A		STA A IFFLG	
0862 7E 08 05		JMP COMAN3	GO DO COMMAND
0865 7F 00 6F	FINCM1	CLR NOFL	CLEAR FLAGS
0868 7F 00 65	FINCM2	CLR CMFLG	
086B 7C 00 BF	FINCM4	INC ENDLIN	SET END LINE
086E 7E 06 8C		JMP GETCHR	GO GET CHARACTER

* CALL MACRO

0871 DF B0	CALMAC	STX XMAC	SAVE POINTER
0873 CE 00 6C		LDX #ATFLG	POINT TO VALUES
0876 A6 00	CALMA2	LDA A 0,X	GET VALUE
0878 36		PSH A	PUT ON STACK
0879 6F 00		CLR 0,X	CLEAR IT
087B 09		DEX	
087C 8C 00 5F		CPX #INC	FINISHED?
087F 26 F5		BNE CALMA2	
0881 96 D4		LDA A CNTFLG	SAVE CNT FLAG
0883 36		PSH A	
0884 7F 00 D4		CLR CNTFLG	
0887 7C 00 8B		INC MACCNT	BUMP COUNTER
0888 DE B0		LDX XMAC	RESTORE COUNT
088C 86 0F		LDA A #\$F	
088E 97 66		STA A MBFLG	SET FLAG
0890 97 BF		STA A ENDLIN	

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 21

0892 EE 02	LDX	2,X	GET ADDRESS
0894 DF 67	STX	MBFPNT	SAVE AS POINTER
0896 7E 03 3D	JMP	PROC	GO PROCESS

* PRINT C. R. AND L. F.

0899 BD 13 ED	PCRLF	JSR	PUSHX	SAVE X
089C C6 18		LDA B	#24	SET COUNT
089E CE 15 00		LDX	#FIXWD+\$EC	
08A1 BD 0B ED		JSR	SENV9	CHECK ACTIVITY
08A4 81 2A		CMP A	#\$2A	OVERFLOW?
08A6 26 CE		BNE	CALMA2	
08A8 8D 07	PCRLF2	BSR	SCRLF	DO CR AND LF
08AA BD 14 71		JSR	TSTBRK	BREAK?
08AD BD 13 FF	PCRLF4	JSR	PULLX	RESTORE X
08B0 39		RTS		RETURN

* SPECIAL CARRIAGE RETURN LINE FEED

08B1 CE 15 43	SCRLF	LDX	#CRLFST	POINT TO STRING
08B4 A6 00	SCRLF1	LDA A	0,X	GET CHARACTER
08B6 81 04		CMP A	#4	IS IT 4?
08B8 27 06		BEQ	SCRLF3	
08B9 BD 14 9F		JSR	OUTCHR	OUTPUT CHAR.
08BD 08		INX		BUMP POINTER
08BE 20 F4		BRA	SCRLF1	
08C0 96 79	SCRLF3	LDA A	NOOUT	DO OUTPUT?
08C2 26 18		BNE	SCRLF4	
08C4 96 94		LDA A	TLPP	LINES PER PAGE?
08C6 27 14		BEQ	SCRLF4	
08C8 7C 00 7A		INC	TOUTL	BUMP LINE COUNT
08CB 91 7A		CMP A	TOUTL	MAX?
08CD 22 0D		BHI	SCRLF4	
08CF 7F 00 7A		CLR	TOUTL	CLEAR COUNT
08D2 BD 02 06		JSR	INCH	WAIT FOR CHARACTER
08D5 81 0D		CMP A	#\$D	IS IT C. R.?
08D7 26 03		BNE	SCRLF4	
08D9 7E 02 09		JMP	MON	EXIT PROCESSOR
08DC 96 91	SCRLF4	LDA A	DIYFLG	DIVERTING?
08DE 26 43		BNE	SCRLF9	
08E0 7C 00 4D		INC	LINCNT	BUMP LINE COUNTER
08E3 96 4D	SCRLF5	LDA A	LINCNT	
08E5 CE 17 D8		LDX	#TRAPS	POINT TO TRAPS
08E8 R1 00	SCRL55	CMP A	0,X	LINE = TRAP?
08ER 27 24		BEQ	SCRLF8	
08EC 08		INX		GET TO NEXT
08ED 08		INX		
08EE 08		INX		
08EF 8C 18 08		CPX	#TRPEND	END?
08F2 26 F4		BNE	SCRL55	
08F4 91 4F	SCRLF6	CMP A	PGL	BOTTOM OF PAGE?
08F6 23 2B		BLS	SCRLF9	
08F8 96 5E		LDA A	NPGN	GET NEW PAGE NUM.
08FA 27 07		BEQ	SCRLF7	

08FC 7F 00 5E		CLR	NPGN	
08FF 97 74		STA A	PGN	SET PAGE NUMBER
0901 20 03		BRA	SCRL75	
0903 7C 00 74	SCRLF7	INC	PGN	BUMP BY ONE
0906 86 01	SCRL75	LDA A	#1	SET UP 1
0908 97 4D		STA A	LINCNT	SET LINE COUNT
090A 96 B5		LDA A	SUPL	CHECK FLAG
090C 26 15		BNE	SCRLF9	
090E 20 D3		BRA	SCRLF5	
0910 7C 00 6C	SCRLF8	INC	ATFLG	BUMP AT COUNT
0913 96 BF		LDA A	ENDLIN	SAVE STATUS
0915 36		PSH A		
0916 A6 01		LDA A	1,X	GET NAME
0918 E6 02		LDA B	2,X	
091A BD 08 15	SCRL85	JSR	COMAN4	GO PROCESS
091D 7A 00 6C		DEC	ATFLG	DEC COUNT
0920 32		PUL A		
0921 97 BF		STA A	ENDLIN	RESTORE STATUS
0923 39	SCRLF9	RTS		RETURN

* BREAK FILLED BUFFER

0924 86 01	BRAK	LDA A	#1	SETUP 1
0926 91 4D		CMP A	LINCNT	TEST LINE COUNT
0928 27 B9		BEQ	SCRLF5	

* FLUSH WORK BUFFER

092A 96 6F	FLUSHB	LDA A	NOFL	NO FLUSH?
092C 26 1F		BNE	FLUSH5	
092E 86 20	FLUSH	LDA A	#\$20	SET UP SPACE
0930 DE DA		LDX	BUFFPNT	SET POINTER
0932 8C 15 E4		CPX	#LINBUF	BEGINNING OF BUFFER?
0935 27 16		BEQ	FLUSH5	
0937 DF R8		STX	STPOUT	SET END
0939 9C DE	FLUSH2	CPX	BUFEND	END?
093B 27 05		BEQ	FLUSH3	
093D A7 00		STA A	0,X	SAVE CHARACTER
093F 08		INX		BUMP POINTER
0940 20 F7		BRA	FLUSH2	
0942 CE 15 E4	FLUSH3	LDX	#LINBUF	POINT TO BUFFER
0945 97 6B		STA A	FLBF	SET FLAG
0947 BD 04 32		JSR	ADJSPC	ADJUST SPACE
094A 7F 00 6B		CLR	FLBF	
094D 7F 00 6F	FLUSH5	CLR	NOFL	CLEAR FLAG
0950 39		RTS		RETURN

* FINISH AND CLEAN UP

0951 8D D7	FINISH	BSR	FLUSHB	FLUSH BUFFER
0953 7C 00 B5	FINIS4	INC	SUPL	
0956 BD 0A 67		JSR	PAGE	GO PAGE
0959 7E 02 09		JMP	NON	EXIT

* COMMAND TABLE

095C 53	CMNDT	FCC	'SP'
095D 50			
095E 0A 44		FDB	SPACE
0960 50		FCC	'PG'
0961 47			
0962 0A 67		FDB	PAGE
0964 4D		FCC	'MS'
0965 53			
0966 0A 8B		FDB	MULTS
0968 53		FCC	'SS'
0969 53			
096A 0A 99		FDB	SNGLS
096C 4E		FCC	'NJ'
096D 4A			
096E 0A 9D		FDB	NOJST
0970 4A		FCC	'JU'
0971 55			
0972 0A A1		FDB	JST
0974 44		FCC	'DH'
0975 48			
0976 0C 04		FDB	DUBH
0978 44		FCC	'DW'
0979 57			
097A 0C 0F		FDB	DUBW
097C 44		FCC	'DB'
097D 42			
097E 0C 18		FDB	DUBB
0980 43		FCC	'CE'
0981 45			
0982 0C 24		FDB	CENTER
0984 42		FCC	'BR'
0985 52			
0986 09 24		FDB	BRAK
0988 2A		FCC	'*' '
0989 20			
098A 0A 66		FDB	SPACE6
098C 46		FCC	'FI'
098D 49			
098E 0C B2		FDB	FILL
0990 4E		FCC	'NF'
0991 46			
0992 0C AB		FDB	NOFILL
0994 53		FCC	'SI'
0995 49			
0996 0C 59		FDB	SIND
0998 50		FCC	'PI'
0999 49			
099A 0C 76		FDB	PTIND
099C 50		FCC	'PN'
099D 4E			
099E 0B F7		FDB	PGNUM
09A0 4C		FCC	'LM'

09A1 4D		
09A2 0A CB	FDB	LEFTM
09A4 49	FCC	'IN'
09A5 4E		
09A6 0A DB	FDB	INDNT
09A8 4C	FCC	'LN'
09A9 4E		
09AA 0A F1	FDB	LENGTH
09AC 4E	FCC	'NS'
09AD 53		
09AE 0B 23	FDB	NOSPC
09B0 52	FCC	'RS'
09B1 53		
09B2 0B 26	FDB	RESPC
09B4 50	FCC	'PL'
09B5 4C		
09B6 0C 44	FDB	PAGEL
09B8 43	FCC	'CP'
09B9 50		
09BA 0C 6D	FDB	STCAP
09BC 4E	FCC	'NC'
09BD 43		
09BE 0C 72	FDB	NOCAP
09C0 4E	FCC	'NL'
09C1 4C		
09C2 0C E6	FDB	NEDL
09C4 53	FCC	'SV'
09C5 56		
09C6 0D 2D	FDB	SAVS
09C8 4F	FCC	'OS'
09C9 53		
09CA 0D 49	FDB	OUTSV
09CC 41	FCC	'RT'
09CD 54		
09CE 0D 52	FDB	RTL
09D0 44	FCC	'DM'
09D1 4D		
09D2 0D 9E	FDB	DEFMAC
09D4 41	FCC	'AM'
09D5 4D		
09D6 0D DC	FDB	APMAC
09D8 52	FCC	'RM'
09D9 4D		
09DA 0D E5	FDB	REMMAC
09DC 44	FCC	'DI'
09DD 49		
09DE 0E 25	FDB	DIVERT
09E0 44	FCC	'DA'
09E1 41		
09E2 0E 44	FDB	DIVRPP
09E4 53	FCC	'ST'
09E5 54		
09E6 0C D2	FDB	STOP
09E8 54	FCC	'TL'

-09E9 4C		
-09EA 0F C1	FDB	TITLE
-09EC 4C	FCC	'LT'
09ED 54		
09EE 0F B4	FDB	TLEN
09F0 43	FCC	'CH'
09F1 48		
-09F2 11 12	FDB	CHNG
09F4 49	FCC	'IF'
09F5 46		
-09F6 10 A5	FDB	IF
09F8 4E	FCC	'NR'
09F9 52		
-09FA 11 65	FDB	NREG
09FC 41	FCC	'AR'
09FD 52		
-09FE 11 84	FDB	ARB
0A00 53	FCC	'SR'
0A01 52		
-0A02 11 88	FDB	SROM
0A04 43	FCC	'CR'
0A05 52		
-0A06 11 8D	FDB	CROM
0A08 41	FCC	'AU'
0A09 55		
-0A0A 11 91	FDB	SAUTO
0A0C 54	FCC	'TC'
0A0D 43		
-0A0E 0B 2A	FDB	TABCH
-0A10 54	FCC	'TF'
0A11 46		
-0A12 0B 35	FDB	TABFIL
0A14 54	FCC	'TA'
0A15 41		
-0A16 0B 41	FDB	STAB
0A18 45	FCC	'EX'
0A19 58		
-0A1A 09 51	FDB	FINISH
0A1C 54	FCC	'TM'
0A1D 4D		
-0A1E 0B 60	FDB	TERM
0A20 47	FCC	'GI'
0A21 49		
-0A22 0B 75	FDB	GETIN
0A24 45	FCC	'EV'
0A25 56		
-0A26 0B 83	FDB	SENV
0A28 52	FCC	'RP'
0A29 50		
-0A2A 0C B9	FDB	RPT
-0A2C 50	FCC	'PS'
0A2D 53		
-0A2E 0A 3D	FDB	PASS
0A30 28	FCC	

0A31 20				
0A32 0A 66		FDB	SPACE6	
0A34 20		FCC	/	/
0A35 20				
0A36 0A 66		FDB	SPACE6	
0A38 20		FCC	/	/
0A39 20				
0A3A 0A 66		FDB	SPACE6	
0A3C 00	TBLEND	FCB	0	

* PASS FILE ROUTINE .PS

0A3D 7F 00 C8	PASS	CLR	FILFLG	FIX FLAGS
0A40 7C 00 8C		INC	PASFLG	
0A43 39		RTS		

* SPACE ROUTINE .SP N

0A44 BD 09 2A	SPACE	JSR	FLUSHB	FLUSH BUFFER
0A47 96 73		LDA A	NSP	NO SPACE?
0A49 26 1B		BNE	SPACE6	
0A4B BD 12 1A		JSR	CHKNUM	CHECK FOR NUMBER
0A4E 96 70		LDA A	INNUM	GET NUMBER
0A50 26 03		BNE	SPACE2	
0A52 7C 00 70		INC	INNUM	INC BY ONE
0A55 BD 0C F5	SPACE2	JSR	FNTR	FIND TRAP
0A58 91 70		CMP A	INNUM	EQUAL?
0A5A 25 02		BLO	SPACE4	
0A5C 96 70		LDA A	INNUM	GET NUMBER
0A5E 36	SPACE4	PSH A		
0A5F BD 08 99		JSR	PCRLF	OUTPUT CR AND LF
0A62 32		PUL A		
0A63 4A		DEC A		DEC COUNT
0A64 26 F8		BNE	SPACE4	
0A66 39	SPACE6	RTS		RETURN

* PAGE ROUTINE .PG +N

0A67 BD 12 1A	PAGE	JSR	CHKNUM	CHECK FOR NUMBER
0A6A 24 07		BCC	PAGE2	
0A6C 96 74		LDA A	PGN	GET PAGE NUMBER
0A6E BD 12 09		JSR	FIXVAL	FIX VALUE
0A71 20 0B		BRA	PAGE4	
0A73 96 73	PAGE2	LDA A	NSP	NO SPACE?
0A75 26 13		BNE	PAGE6	
0A77 96 5E		LDA A	NPGN	GET NEW PAGE NUM.
0A79 26 03		BNE	PAGE4	
0A7B 96 74		LDA A	PGN	
0A7D 4C		INC A		BUMP BY ONE
0A7E 97 5E	PAGE4	STA A	NPGN	SAVE AS NEW
0A80 BD 09 2A		JSR	FLUSHB	FLUSH BUFFER
0A83 BD 08 99	PAGE5	JSR	PCRLF	OUTPUT CR & LF
0A86 96 5E		LDA A	NPGN	GET NEW PAGE NUM.
0A88 26 F9		BNE	PAGES	

SC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 27

0A8A 39 PAGE6 RTS RETURN

* MULTIPLE SPACE ROUTINE . MS +N

0A8B BD 12 1A	MULTS	JSR	CHKNUM	CHECK FOR NUMBER
0A8E 24 04		BCC	MULTS2	
0A90 96 70		LDA A	INNUM	GET NUMBER
0A92 20 02		BRA	MULTS3	
0A94 86 02	MULTS2	LDA A	#2	DEFAULT IS 2
0A96 97 D2	MULTS3	STA A	MSC	SET COUNT
0A98 39		RTS		

* SINGLE SPACE ROUTINE . SS

0A99 7F 00 D2	SNGLS	CLR	MSC	CLEAR COUNT
0A9C 39		RTS		

* NO ADJUST ROUTINE . NJ

0A9D 7F 00 D6	NOJST	CLR	JUST	CLEAR JUST FLAG
0AA0 39		RTS		

* SET JUSTIFICATION ROUTINE . JU C

0AA1 97 D6	JST	STA A	JUST	SET FLAG
0AA3 BD 11 D8		JSR	LDNSKP	GET NEXT CHAR.
0AA6 BD 11 E6		JSR	CLSFY	CLASSIFY IT
0AA9 C1 02		CMP B	#2	
0AAB 26 09		BNE	JST15	
0AAD 81 4E		CMP A	#'N	NORMAL?
0AAF 26 06		BNE	JST2	
0AB1 4F	JST1	CLR A		ADJUST FLAGS
0AB2 97 CE		STA A	CNJ	
0AB4 97 D0		STA A	RTJ	
0AB6 39	JST15	RTS		RETURN
0AB7 81 52	JST2	CMP A	#'R	RIGHT HAND?
0AB9 26 06		BNE	JST3	
0ABB 7F 00 CE		CLR	CNJ	FIX FLAGS
0ABE 97 D0		STA A	RTJ	
0AC0 39		RTS		
0AC1 81 43	JST3	CMP A	#'C	CENTERED?
0AC3 26 EC		BNE	JST1	
0AC5 7F 00 D0		CLR	RTJ	FIX FLAGS
0AC8 97 CE		STA A	CNJ	
0ACA 39	JST4	RTS		RETURN

* SET LEFT MARGIN . LM +N

0ACB BD 12 1A	LEFTM	JSR	CHKNUM	CHECK FOR NUMBER
0ACE 24 0A		BCC	LEFTM2	
0AD0 96 4E		LDA A	LM	GET MARGIN
0AD2 BD 12 09		JSR	FIXVAL	FIX VALUE
0AD5 2A 01		BPL	LEFTM1	
0AD7 4F		CLR A		

0AD8 97 4E LEFTM1 STA A LFM SET NEW VALUE
 0ADR 39 LEFTM2 RTS RETURN

* SET INDENT .IN +N

0ADB BD 09 2A	INDNT	JSR	FLUSHB	FLUSH BUFFER
0ADE BD 12 1A		JSR	CHKNUM	CHECK FOR NUMBER
0AE1 24 F7		BCC	LEFTM2	
0AE3 96 48		LDA A	IND	GET INDENT
0AE5 BD 12 09		JSR	FIXVAL	FIX VALUE
0AE8 2A 01		BPL	INDNT2	
0AEA 4F		CLR A		
0AEB 98 48	INDNT2	SUB A	IND	SET INDENT
0AED 97 B3		STA A	TIND	SAVE AS TEMP
0REF 28 14		BRA	LENT25	

* SET LENGTH OF LINE .LN +N

0AF1 BD 12 1A	LENTH	JSR	CHKNUM	CHECK FOR NUMBER
0AF4 24 19		BCC	LENTH5	
0AF6 96 4B		LDA A	LLN	GET LENGTH 35
0AF8 BD 12 09		JSR	FIXVAL	FIX VALUE
0AFB 81 0E		CMP A	#14	14 OR LESS?
0AFD 22 02		BHI	LENTH2	
0AFF 86 0F		LDA A	#15	FORCE TO 15
0B01 90 48	LENTH2	SUB A	LLN	SET NEW
0B03 97 B4		STA A	TLLN	SAVE AS TEMP 10
0B05 DE DA	LENT25	LDX	BUFPNT	CHECK POINTER
0B07 8C 15 E4		CPX	#LINBUF	
0B0A 26 03		BNE	LENTH5	
0B0C 7E 14 14		JMP	FIXWD	GO FIX WIDTH
0B0F 39	LENTH5	RTS		RETURN

* DO NECESSARY TABMING

0B10 D6 42	DOTAB	LDA B	COLCNT	GET COUNT
0B12 DE 80		LDX	NXTTAB	POINT TO TAB
0B14 E1 00		CMP B	0,X	COMPARE
0B16 24 05		BHS	DOTAB2	
0B18 96 C1		LDA A	TFILL	GET FILL CHAR.
0B1A 7E 07 0A		JMP	FETCH3	
0B1D 7F 00 82	DOTAB2	CLR	TABFLG	CLEAR FLAG
0B20 7E 06 8C		JMP	GETCHR	BACK TO GET CHAR.

* SET NO SPACE .NS

0B23 97 73	NOSPC	STA A	NSP	SET FLAG
0B25 39		RTS		

* RESTORE SPACE MODE .RS

0B26 7F 00 73	RESPC	CLR	NSP	CLEAR FLAG
0B29 39		RTS		

* DEFINE TAB CHARACTER . TC C

0B2A BD 11 D8	TABCH	JSR	LDNSKP	GET TO NEXT CHAR.
0B2D 81 0D		CMP A	#\$D	IS IT A C. R. ?
0B2F 26 01		BNE	TABCH2	
0B31 4F		CLR A		CLEAR VALUE
0B32 97 C8	TABCH2	STA A	TRB	SAVE TAB CHAR.
0B34 39		RTS		RETURN

* DEFINE TAB FILL CHARACTER . TF C

0B35 BD 11 D8	TABFIL	JSR	LDNSKP	GET TO NEXT CHAR.
0B38 81 0D		CMP A	#\$D	IS IT C. R. ?
0B3A 26 02		BNE	TABFI2	
0B3C 86 A0		LDA A	#\$A0	SET UNPAD SPACE
0B3E 97 C1	TABFI2	STA A	TFILL	SAVE CHAR.
0B40 39		RTS		RETURN

* DEFINE TAB COLUMNS . TR 1 2 3 4

0B41 CE 01 10	STAB	LDX	#TBBS	POINT TO TBBS
0B44 BD 13 ED	STAB2	JSR	PUSHX	SAVE X
0B47 BD 12 1A		JSR	CHKNUM	CHECK FOR NUMBER
0B4A 24 0E		BCC	STAB4	
0B4C BD 13 FF		JSR	PULLX	RESTORE
0B4F 96 70		LDA A	INNUM	GET NUMBER
0B51 A7 00		STA A	0,X	SAVE IT
0B53 08		INX		BUMP POINTER
0B54 8C 01 24		CPX	#TABEND	END OF TABLE?
0B57 26 EB		BNE	STAB2	
0B59 39		RTS		RETURN
0B5A BD 13 FF	STAB4	JSR	PULLX	
0B5D 6F 00		CLR	0,X	CLEAR LAST
0B5F 39		RTS		

* OUTPUT STRING TO TERMINAL . TM STRING

0B60 BD 11 D8	TERM	JSR	LDNSKP	GET TO NEXT CHAR.
0B63 A6 00	TERM2	LDA A	0,X	GET CHAR.
0B65 81 0D		CMP A	#\$D	IS IT C. R. ?
0B67 27 03		BEQ	TERM4	
0B69 08		INX		BUMP THE ROPINTER
0B6A 20 F7		BRA	TERM2	
0B6C 86 04	TERM4	LDA A	#4	SET UP 4
0B6E A7 00		STA A	0,X	SAVE IT
0B70 DE E6		LDX	CMNPNT	SET POINTER
0B72 7E 14 90		JMP	PSTRNG	GO PRINT STRING

* GET INPUT FROM TERMINAL . GI PROMPT

0B75 8D E9	GETIN	BSR	TERM	GO PRINT PROMPT
0B77 CE 15 62		LDX	#QUSTR	POINT TO STR.
0B7A BD 14 92		JSR	PDATA	OUTPUT IT
0B7D BD 14 44		JSR	GIBUF	GET INPUT RESPONSE

0B80 97 9E STA A SBFLG SET FLAG
 0B82 39 RTS RETURN

* SET NEW ENVIRONMENT .EV N

0B83 BD 12 1A	SENV	JSR	CHKNUM	CHECK FOR NUMBER
0B86 24 08		BCC	SENV1	
0B88 96 70		LDA R	INNUM	GET NUMBER
0B8A 27 05		BEQ	SENV2	
0B8C 86 01		LDA R	#1	SET UP 1
0B8E 20 01		BRA	SENV2	
0B90 4F	SENV1	CLR A		CLEAR VALUE
0B91 91 7E	SENV2	CMP R	EV	PRESENT EV?
0B93 26 01		BNE	SENV3	
0B95 39		RTS		YES, RETURN
0B96 97 7E	SENV3	STA A	EV	SET NEW EV
0B98 96 42		LDA R	COLCNT	SAVE COL COUNT
0B9A D6 83		LDA B	COLCN2	
0B9C D7 42		STA B	COLCNT	
0B9E 97 83		STA R	COLCN2	
0BA0 96 48		LDA R	IND	FIX THE INDENT
0BA2 D6 84		LDA B	IND2	
0BA4 97 84		STA R	IND2	
0BA6 D7 48		STA B	IND	
0BA8 96 4B		LDA R	LLN	DO LINE LENGTH
0BAA D6 9F		LDA B	LLN2	
0BAC 97 9F		STA R	LLN2	
0BAE D7 4B		STA B	LLN	
0BB0 CE 00 C2	SENV4	LDX	#AUTO	POINT TO BLOCK
0BB3 A6 00		LDA R	0,X	GET VALUE
0BB5 E6 01		LDA B	1,X	
0BB7 A7 01		STA R	1,X	SWAP VALUE
0BB9 E7 00		STA B	0,X	
0BBB 08		INX		GO TO NEXT
0BBC 08		INX		
0BBD 8C 00 DA		CPX	#BUFPNT	FINISHED?
0BC0 26 F1		BNE	SENV4	
0BC2 A6 00	SENV6	LDA R	0,X	GET VALUE
0BC4 E6 02		LDA B	2,X	
0BC6 A7 02		STA R	2,X	SWAP
0BC8 E7 00		STA B	0,X	
0BCA A6 01		LDA R	1,X	
0BCC E6 03		LDA B	3,X	
0BCE A7 03		STA R	3,X	
0BD0 E7 01		STA B	1,X	
0BD2 08		INX		BUMP THE POINTER
0BD3 08		INX		
0BD4 08		INX		
0BD5 08		INX		
0BD6 8C 00 E6		CPX	#CMNPNT	FINISHED?
0BD9 26 E7		BNE	SENV6	
0BDB CE 15 E4		LDX	#LINBUF	POINT TO BUFFER
0BDE A6 00	SENV8	LDA R	0,X	GET A CHAR.
0BE0 E6 C8		LDA B	200,X	

0BE2 A7 C8		STA A	200,X	SWAP FOR NEW
0BE4 E7 00		STA B	0,X	
0BE6 08		INX		BUMP TO NEXT
0BE7 8C 16 AC		CPX	#LINBU2	FINISHED?
0BEA 26 F2		BNE	SENV8	
0BEC 39		RTS		RETURN
0BED A6 00	SENV9	LDA A	0,X	GET VALUE
0BEF 08		INX		BUMP TO NEXT
0BF0 AB 00	SENV95	ADD A	0,X	ADD IN BIAS
0BF2 08		INX		
0BF3 5A		DEC B		DEC THE VALUE
0BF4 26 FA		BNE	SENV95	
0BF6 39		RTS		RETURN

* SET NEW PAGE NUMBER .PN +N

0BF7 BD 12 1A	PGNUM	JSR	CHKNUM	CHECK FOR NUMBER
0BFA 24 07		BCC	PGNUM4	
0BFC 96 74		LDA A	PGN	GET VALUE
0BFE BD 12 09		JSR	FIXVAL	GO FIX VALUE
0C01 97 74		STA A	PGN	SAVE NEW
0C03 39	PGNUM4	RTS		RETURN

* SET DOUBLE HEIGHT .DH

0C04 BD 09 2E	DUBH	JSR	FLUSH	FLUSH BUFFER
0C07 86 12	DUBH1	LDA A	#\$12	SET UP CODE
0C09 7C 00 4D		INC	LINCNT	BUMP LINE COUNT
0C0C 7E 14 9F	DUBH2	JMP	OUTCHR	OUTPUT CHARACTER

* SET DOUBLE WIDTH .DW

0C0F BD 09 2E	DUBW	JSR	FLUSH	FLUSH BUFFER
0C12 86 0E		LDA A	#\$0E	SET UP CODE
0C14 97 8E		STA A	DWFLG	SET FLAG
0C16 20 F4		BRA	DUBH2	

* SET DOUBLE BOTH .DB

0C18 BD 09 2E	DUBB	JSR	FLUSH	FLUSH BUFFER
0C1B 86 0E		LDA A	#\$0E	SET UP CODE
0C1D 97 8E		STA A	DWFLG	SET FLAG
0C1F BD 14 9F		JSR	OUTCHR	OUTPUT CHARACTER
0C22 20 E3		BRA	DUBH1	

* CENTER N LINES .CE +N

0C24 BD 09 2A	CENTER	JSR	FLUSHB	FLUSH BUFFER
0C27 BD 12 1A		JSR	CHKNUM	CHECK FOR NUMBER
0C29 24 0B		BCC	CENTE2	
0C2C 96 D4		LDA A	CNTFLG	GET OLD COUNT
0C2E BD 12 09		JSR	FIXVAL	FIX VALUE
0C31 97 D4		STA A	CNTFLG	SAVE NEW
0C33 27 23		SEQ	PAGE14	

0C35 20 04		BRA	CENTE4	
0C37 86 01	CENTE2	LDA A	#1	DEFAULT TO 1
0C39 97 D4		STA A	CNTFLG	SAVE COUNT
0C3B 96 C8	CENTE4	LDA A	FILFLG	GET FLAG
0C3D 97 6E		STA A	TFILF	SAVE AS TEMP
0C3F 86 FF		LDA A	#\$FF	
0C41 97 C8		STA A	FILFLG	FORCE FILL MODE
0C43 39		RTS		RETURN

* SET PAGE LENGTH . PL +N

0C44 BD 12 1A	PAGEL	JSR	CHKNUM	CHECK FOR NUMBER
0C47 25 04		BCS	PAGEL1	
0C49 86 42		LDA A	#66	DEFAULT TO 66
0C4B 20 09		BRA	PAGEL2	
0C4D 96 4F	PAGEL1	LDA A	PGL	GET LAST VALUE
0C4F BD 12 09		JSR	FIXVAL	FIX VALUE
0C52 14 D		TST A		
0C53 26 01		BNE	PAGEL2	
0C55 4C		INC A		BUMP BY ONE
0C56 97 4F	PAGEL2	STA A	PGL	SAVE NEW
0C58 39	PAGEL4	RTS		RETURN

* SET SINGLE INDENT . SI +N

0C59 BD 09 2A	SIND	JSR	FLUSHB	FLUSH BUFFER
0C5C BD 12 1A		JSR	CHKNUM	CHECK FOR NUMBER
0C5F 24 F7		BCC	PAGEL4	
0C61 96 7C		LDA A	SIN	GET OLD VALUE
0C63 BD 12 09		JSR	FIXVAL	GO FIX VALUE
0C66 90 7C		SUB A	SIN	
0C68 97 B2		STA A	TSIN	SAVE AS TEMP
0C6A 7E 0B 05		JMP	LENT25	

* SET CAPS MODE . CP

0C6D 86 0F	STCAP	LDA A	#\$F	SET FLAG
0C6F 97 77		STA A	DOCAP	
0C71 39		RTS		

* CLEAR CAPS MODE . NC

0C72 7F 00 77	NOCAP	CLR	DOCAP	CLEAR FLAG
0C75 39		RTS		

* PUT IN INDENT FIELD . PI STRING

0C76 BD 09 2A	PTIND	JSR	FLUSHB	FLUSH BUFFER
0C79 BD 11 D8		JSR	LDNSKP	GET TO NEXT CHAR.
0C7C D6 48		LDA B	IND	GET INDENT
0C7E 27 2A		BEN	PTIND5	
0C80 D7 69		STA B	NOCR	SET FLAG
0C82 5F		CLR B		
0C83 37	PTIND2	PSH B		

0C84 BD 06 8C		JSR	GETCHR	GO GET CHAR.
0C87 33		PUL B		
0C88 81 0D		CMP R	#\$D	CHECK IF C.R.?
0C8A 27 0C		BEQ	PTIND3	
0C8C 37		PSH B		
0C8D BD 14 9F		JSR	DUTCHR	GO OUTPUT CHAR.
0C90 33		PUL B		
0C91 5C		INC B		BUMP COUNT
0C92 D1 48		CMP B	IND	FINISHED?
0C94 24 0E		BHS	PTIND4	
0C96 20 EB		BRA	PTIND2	
0C98 86 20	PTIND3	LDA A	#\$20	SET UP SPACE
0C9A 37		PSH B		
0C9B BD 14 9F		JSR	DUTCHR	OUTPUT IT
0C9E 33		PUL B		
0C9F 5C		INC B		BUMP COUNT
0CA0 D1 48		CMP B	IND	FINISHED?
0CA2 25 F4		BLO	PTIND3	
0CA4 5C	PTIND4	INC B		BUMP COUNT
0CA5 D7 7B		STA B	PTFL	SET FLAG
0CA7 7F 00 69		CLR	NOCR	
0CAA 39	PTIND5	RTS		RETURN

* SET NOFILL MODE .NF

0CB8 BD 09 2A	NOFILL	JSR	FLUSHB	FLUSH BUFFER
0CAE 7F 00 C8		CLR	FILFLG	CLEAR FLAG
0CB1 39		RTS		

* SET FILL MODE .FI

0CB2 BD 09 2A	FILL	JSR	FLUSHB	FLUSH BUFFER
0CB5 7C 00 C8		INC	FILFLG	SET FLAG
0CB8 39		RTS		

* REPEAT COMMAND .RP

0CB9 BD 09 2A	RPT	JSR	FLUSHB	FLUSH BUFFER
0CBC 7C 00 B5		INC	SUPL	SET FLAG
0CBF BD 0A 67		JSR	PAGE	GO PAGE
0CC2 BD 14 E5		JSR	RWND	REWIND FILE
0CC5 EE 01 FF		LDS	#STACK	*** STACK REF. ***
0CC8 4F		CLR A		
0CC9 CE 00 5A		LDX	#GDNUM	CLEAR SPACE
0CCC BD 02 AC		JSR	CLRSP2	
0CCF 7E 02 C0		JMP	INIT	GO INIT.

* STOP COMMAND .ST

0CD2 BD 09 2A	STOP	JSR	FLUSHB	FLUSH BUFFER
0CD5 CE 15 58		LDX	#STPSTR	POINT TO STRING
0CD8 BD 14 90	STOP1	JSR	PSTRNG	OUTPUT IT
0CDB BD 02 06		JSR	INCH	GO GET CHAR.
0CDE 81 53		CMP R	#'S'	IS IT 'S'?

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 34

0CE0 26 03		BNE	STOP2	
0CE2 7E 09 53		JMP	FINIS4	GO FINISH
0CE5 39	STOP2	RTS		RETURN

* NEED N LINES . NL N

0CE6 BD 12 1A	NEDL	JSR	CHKNUM	CHECK FOR NUMBER
0CE9 25 03		BCS	NEDL1	
0CEB 7C 00 70		INC	INNUM	BUMP BY 1
0CEE 8D 05	NEDL1	BSR	FNTR	GO FIND TRAP
0CF0 91 70	NEDL2	CMP A	INNUM	COMPARE
0CF2 25 4A		BLO	SAVS25	
0CF4 39	NEDL4	RTS		

* FIND THE NEXT TRAP

0CF5 86 FF	FNTR	LDA A	#\$FF	SET MIN DISTANCE
0CF7 97 7D		STA A	MINDIS	
0CF9 CE 17 D8		LDX	#TRAPS	POINT TO TRAPS
0CFc D6 4D	FNTR2	LDA B	LINCNT	GET COUNT
0CFE E1 00		CMP B	0, X	CHECK LOC.
0D00 24 0F		BHS	FNTR4	
0D02 A6 00		LDA A	0, X	GET DISTANCE
0D04 81 FF		CMP A	#\$FF	
0D06 27 11		BEQ	FNTR5	
0D08 10		SBA		SWAP REGISTERS
0D09 91 7D		CMP A	MINDIS	MIN DISTANCE?
0D0B 24 04		BHS	FNTR4	
0D0D 97 7D		STA A	MINDIS	SAVE NEW
0D0F DF 85		STX	NXTTRP	SAVE POINTER
0D11 08	FNTR4	INX		BUMP THE POINTER
0D12 08		INX		
0D13 08		INX		
0D14 8C 18 08		CPX	#TRPEND	FINISHED?
0D17 26 E3		BNE	FNTR2	
0D19 D6 7D	FNTR5	LDA B	MINDIS	GET DISTANCE
0D1B C1 FF		CMP B	#\$FF	
0D1D 26 07		BNE	FNTR6	
0D1F 96 4F		LDA A	PGL	SET UP PAGE LEN.
0D21 90 4D		SUB A	LINCNT	
0D23 4C		INC A		FIX VALUE
0D24 5F		CLR B		
0D25 39		RTS		RETURN
0D26 E6 00	FNTR6	LDA B	0, X	
0D28 96 7D		LDA A	MINDIS	GET DISTANCE
0D2A DE 85		LDX	NXTTRP	POINT TO TRAP
0D2C 39		RTS		RETURN

* SAVE SPACE ROUTINE . SV N

0D2D 7F 00 87	SAVS	CLR	SVDSPC	CLEAR COUNT
0D30 BD 12 1A		JSR	CHKNUM	CHECK FOR NUMBER
0D33 25 03		BCS	SAVS1	
0D35 7C 00 70		INC	INNUM	GET COUNT

0D38 BD BB	SAVS1	BSR	FNTR	FIND TRAP
0D3A 91 70	SAVS2	CMP A	INNUM	
0D3C 25 86		BLO	SAVS4	
0D3E 7F 00 73	SAVS25	CLR	NSP	CLEAR NO SPACE
0D41 7E 0A 5E		JMP	SPACE4	GO DO SPACE
0D44 96 70	SAVS4	LDA A	INNUM	GET COUNT
0D46 97 87		STA A	SVDSPEC	SAVE COUNT
0D48 39	SAVS5	RTS		RETURN

* OUTPUT SAVED SPACE . OS

0D49 96 87	TUTSY	LDA R	SVDSPEC	GET REMEMBERED COUNT
0D4B 27 FB		BEQ	SAVS5	
0D4D 7F 00 87		CLR	SVDSPEC	CLEAR COUNT
0D50 20 EC		BRA	SAVS25	OUTPUT SPACE

* AT LINE N ROUTINE . AT -N

0D52 BD 12 1A	ATL	JSR	CHKNUM	CHECK FOR NUMBER
0D55 24 28		BCC	ATL35	
0D57 BD 11 06		JSR	TSTNEG	IS IT NEGATIVE?
0D5A BD 0F 8A		JSR	GTNAME	GET NAME
0D5D 96 4F		LDA A	PGL	GET PAGE LEN.
0D5F 4C		INC A		
0D60 BD 12 09		JSR	FIXVAL	FIX THE VALUE
0D63 4D		TST A		
0D64 26 01		BNE	ATL1	
0D66 4C		INC A		BUMP BY ONE
0D67 CE 17 D8	ATL1	LDX	#TRAPS	POINT TO TRAPS
0D6A A1 00	ATL2	CMP A	0,X	COMPARE
0D6C 27 12		BEQ	ATL4	
0D6E 8D 27		BSR	INTRP	
0D70 26 F8		BNE	ATL2	
0D72 CE 17 D8		LDX	#TRAPS	POINT TO TRAPS
0D75 C6 FF		LDA B	#\$FF	SET REFERENCE
0D77 E1 00	ATL3	CMP B	0,X	
0D79 27 14		BEQ	ATL5	
0D7B 8D 1A		BSR	INTRP	
0D7D 26 F8		BNE	ATL3	
0D7F 39		RTS		RETURN
0D80 D6 A0	ATL4	LDA B	MACNAME	GET NAME
0D82 26 04		BNE	ATL45	
0D84 5A		DEC B		DEC THE COUNT
0D85 E7 00		STA B	0,X	SAVE POSITION
0D87 39		RTS		RETURN
0D88 96 A1	ATL45	LDA A	MACNAME+1	GET NAME
0D8A E7 01		STR B	1,X	SAVE CHAR.
0D8C A7 02		STR A	2,X	
0D8E 39		RTS		RETURN
0D8F D6 A0	ATL5	LDA B	MACNAME	GET NAME
0D91 27 EC		BEQ	ATL35	
0D93 A7 00		STA A	0,X	SAVE CHARACTER
0D95 20 F1		BRA	ATL45	

* INCREMENT TRAP POINTER

0D97 08	INTRP	INX	FIX POINTER
0D98 08		INX	
0D99 08		INX	
0D9A 8C 18 08		Cpx	*TRPEND FINISHED?
0D9D 39		RTS	

* DEFINE MACRO

0D9E 96 66	DEFMRC	LDA A	MBFLG	CHECK DEF FLAG
0DA0 26 39		BNE	DEFMAS	
0DA2 BD 0E 4F		JSR	OPMAC	GO OPEN MACRO
0DA5 27 34	DEFMA2	BEQ	DEFMAS	
0DA7 7C 00 8F		INC	DFMFLG	SET DEF FLAG
0DAA 7F 00 65		CLR	CMFLG	CLEAR COMMAND
0DAD BD 06 89	DEFMA3	JSR	CLRGET	GO GET CHARACTER
0DB0 81 2E		CMP A	#'	IS IT A PERIOD?
0DB2 26 0E		BNE	DEFM35	
0DB4 BD 06 89		JSR	CLRGET	GET NEXT CHAR.
0DB7 81 2E		CMP A	#'	IS IT A PERIOD?
0DB9 27 13		BEQ	DEFMA4	
0DBB 36		PSH A	'`	SAVE CHAR
0DBC 86 2E		LDA A	#'	SET UP PERIOD
0DBE BD 0E RE		JSR	OUTMAC	OUTPUT TO MACRO
0DC1 32		PUL A		RESTORE CHAR
0DC2 BD 0E RE	DEFM35	JSR	OUTMAC	OUTPUT TO MACRO
0DC5 81 BD		CMP A	#\$D	IS IT A C.R.?
0DC7 27 E4		BEQ	DEFMA3	
0DC9 BD 06 89		JSR	CLRGET	GET NEXT CHAR.
0DCC 20 F4		BRA	DEFM35	
0DCE BD 0E F8	DEFMA4	JSR	CLSMAC	CLOSE MACRO
0DD1 BD 06 89	DEFM45	JSR	CLRGET	GET CHARACTER
0DD4 81 BD		CMP A	#\$D	IS IT A C.R.?
0DD6 26 F9		BNE	DEFM45	
0DD8 7F 00 8F		CLR	DFMFLG	CLEAR DEF FLAG
0DDE 39	DEFMAS	RTS		RETURN

* APPEND TO A MACRO .AP XX

0DDC 96 66	RPMAC	LDA A	MBFLG	CHECK FLAG
0DDE 26 FB		BNE	DEFMAS	
0DE0 BD 0E 82		JSR	OPAPP	OPEN FOR APPEND
0DE3 20 C0		BRA	DEFMA2	

* REMOVE MACRO .RM XX

0DE5 BD 0F 8A	REMMAC	JSR	GTNAM	GO GET NAME
0DE8 BD 0F 35		JSR	FNDMAC	FIND MACRO
0DEB 27 01		BEQ	REMMA4	
0DED 39		RTS		RETURN
0DEE DF A2	REMMMA4	STX	MACTMP	SAVE POINTER
0DF0 EE 02		LDX	2,X	GET ADDRESS
0DF2 BD 0F 64		JSR	CHKLST	LAST MACRO?

0DF5 24 14		BCC	REMMA6	
0DF7 DE A2		LDX	MACTMP	GET POINTER
BDF9 A6 02		LDA A	2,X	GET ADDRESS
0DFB E6 03		LDA B	3,X	
0DFD DE A4		LDX	LSTAVL	SET LAST AVAIL
0DFF R7 00		STA A	0,X	
0E01 E7 01		STA B	1,X	
0E03 DE AC		LDX	NXTMAC	SET UP NXT MAC
0E05 DF A4		STX	LSTAVL	SAVE AS LAST AVAIL
0E07 DE A2		LDX	MACTMP	
0E09 20 0A		BRA	REMNAME	
0E0B DE A2	REMMA6	LDX	MACTMP	SET UP POINTER
0E0D A6 02		LDA A	2,X	GET ADDRESS
0E0F E6 03		LDA B	3,X	
0E11 97 A6		STA A	FSTAVL	SET FIRST AVAIL
0E13 D7 A7		STA B	FSTAVL+1	

* REMOVE MACRO NAME FROM TABLE

0E15 E6 04	REMNAME	LDR B	4,X	MOVE CHAR DOWN
0E17 E7 00		STA B	0,X	
0E19 08		INX		BUMP THE POINTER
0E1A 9C FR		CPX	MACEND	FINISHED?
0E1C 26 F7		BNE	REMNAME	
0E1E 09		DEX		DEC THE POINTER
0E1F 09		DEX		
0E20 09		DEX		
0E21 09		DEX		
0E22 DF FA		STX	MACEND	SET NEW END
0E24 39		RTS		RETURN

* DIVERT .DI XX

0E25 96 91	DIVERT	LDA A	DIVFLG	CHECK DIV FLAG
0E27 27 0D		BEQ	DIVER2	
0E29 7C 00 92	DIVER0	INC	DIVFL2	SET MARKER
0E2C 7E 0E F8		JMP	CLSMAC	CLOSE MACRO
0E2F 7F 00 91	DIVER1	CLR	DIVFLG	CLEAR FLAGS
0E32 7F 00 92		CLR	DIVFL2	
0E35 39		RTS		RETURN
0E36 7C 00 92	DIVER2	INC	DIVFL2	SET MARKER
0E39 8D 14		BSR	OPMAC	GO OPEN MACRO
0E3B 27 F2	DIVER4	BEP	DIVER1	
0E3D 7C 00 91		INC	DIVFLG	SET FLAG
0E40 7F 00 55		CLR	LDIV	CLEAR COUNT
0E43 39		RTS		RETURN

* DIVERT APPEND .DA XX

0E44 96 91	DIVAPP	LDA A	DIVFLG	CHECK DIV FLAG
0E46 26 E1		BNE	DIVER0	
0E48 7C 00 92		INC	DIVFL2	SET MARKER
0E4B 8D 35		BSR	OPAPP	OPEN FOR APPEND
0E4D 20 EC		BRA	DIVER4	

* OPEN A MACRO SPACE

0E4F BD 0F 8A	DPMAC	JSR	GTNAM	GET MACRO NAME
0E52 96 A0		LDA A	MACNAM	
0E54 26 01		BNE	OPMAC2	PRESENT?
0E56 39		RTS		
0E57 BD 0F 35	OPMAC2	JSR	FNDMAC	LOOK FOR MACRO
0E5A 26 04		BNE	DPMAC4	
0E5C 8D 90		BSR	REMMA4	REMOVE OLD VERSION
0E5E 20 F7		BRA	OPMAC2	OPEN MACRO
0E60 96 A0	OPMAC4	LDA A	MACNAM	GET NAME
0E62 D6 A1		LDA B	MACNAM+1	
0E64 8C 19 F2		CPX	#MTEND	END OF TABLE?
0E67 26 03		BNE	OPMAC5	
0E69 7E 0E E0		JMP	SYSERR	REPORT ERROR
0E6C A7 00	OPMAC5	STA A	0,X	SAVE NAME
0E6E E7 01		STA B	1,X	
0E70 96 A6		LDA A	FSTAVL	GET FIRST AVAIL
0E72 D6 A7		LDA B	FSTAVL+1	
0E74 A7 02		STA A	2,X	SAVE IN TABLE
0E76 E7 03		STA B	3,X	
0E78 08		INX		BUMP THE POINTER
0E79 08		INX		
0E7A 08		INX		
0E7B 08		INX		
0E7C DF FA		STX	MACEND	SET NEW END
0E7E DE A6		LDX	FSTAVL	GET LAST AVAIL
0E80 20 1E		BRA	SAVSX	

* OPEN MACRO FOR APPEND

0E82 BD 0F 8A	OPAPP	JSR	GTNAM	GET MACRO NAME
0E85 96 A0		LDA A	MACNAM	
0E87 26 01		BNE	OPAPP2	
0E89 39		RTS		NO NAME
0E8A BD 0F 35	OPAPP2	JSR	FNDMAC	FIND MACRO
0E8D 26 D1		BNE	OPMAC4	
0E8F EE 02	OPAPP4	LDX	2,X	GET LOCATION
0E91 BD 0F 64		JSR	CHKLST	IS IT THE LAST ONE?
0E94 24 0A		BCC	SAVSX	
0E96 96 A6		LDA A	FSTAVL	GET FIRST AVAIL
0E98 D6 A7		LDA B	FSTAVL+1	
0E9A A7 00		STA A	0,X	SET NEW
0E9C E7 01		STA B	1,X	
0E9E DE A6		LDX	FSTAVL	

* SAVE SPECIAL INDEX

0EA0 7D 00 92	SAVSX	TST	DIYFL2	TEST MARKER
0EA3 27 06		BEQ	SAVSX2	
0EA5 7F 00 92		CLR	DIYFL2	CLEAR MARKER
0EA8 DF AE		STX	NXTOUT	SAVE POINTER
0EAA 39		RTS		RETURN

0EBB DF AC SAVSX2 STX NXTMAC SAVE POINTER
 0EAD 39 RTS

* OUTPUT TO MACRO SPACE

0EAE DF B0	OUTMAC	STX	XMAC	SAVE POINTER
0EB0 7D 00 92		TST	DIVFL2	TEST MARKER
0EB3 27 04		BEQ	OUTMA0	
0EB5 DE AE		LDX	NXTOUT	SET POINTER
0EB7 20 02		BRA	OUTMA1	
0EB9 DE AC	OUTMA0	LDX	NXTMAC	
0EBB 6D 00	OUTMA1	TST	0,X	TEST IF END
0EBD 27 18		BEQ	OUTMA4	
0EBF 81 1F		CMP A	#\$1F	IS IT \$1F?
0EC1 22 0C		BHI	OUTM18	
0EC3 81 0D		CMP A	#\$D	IS IT C.R.?
0EC5 26 0B		BNE	OUTMR3	
0EC7 7D 00 92		TST	DIVFL2	TEST MARKER
0ECA 27 03		BEQ	OUTM18	
0ECC 7C 00 55		INC	LDIV	BUMP DIV LINE CNT
0ECF A7 00	OUTM18	STR A	0,X	PUT CHARACTER
0ED1 08	OUTMR2	INX		BUMP THE POINTER
0ED2 8D CC	OUTMR3	BSR	SAVSX	GO SAVE X
0ED4 DE B0		LDX	XMAC	RESTORE POINTER
0ED6 39		RTS		
0ED7 08	OUTMA4	INX		BUMP THE POINTER
0ED8 9C A4		CPX	LSTAVL	LAST AVAIL?
0EDA 27 04		BEQ	SYSERR	ERROR?
0EDC EE 00		LDX	0,X	GET POINTER
0EDF 20 DB		BRA	OUTMA1	

* REPORT SYSTEM MACRO ERROR

0EE0 7E 08 4E SYSERR JMP MACOVF REPORT OVERFLOW

* INPUT TO MACRO SPACE

0EE3 DE 67	INMAC	LDX	MBFPNT	SET UP POINTER
0EE5 A6 00	INMAC2	LDA A	0,X	GET THE CHARACTER
0EE7 08		INX		BUMP THE POINTER
0EE8 DF 67		STX	MBFPNT	SAVE IT
0EEA 4D		TST A		TEST THE CHAR.
0EEB 26 06		BNE	INMAC4	
0EED EE 00		LDX	0,X	GET LINK
0EEF 26 F4		BNE	INMAC2	
0EF1 20 04		BRA	INMAC5	
0EF3 81 FF	INMAC4	CMP A	#\$FF	IS CHAR FF?
0EF5 27 EE		BEQ	INMAC2	
0EF7 39	INMAC5	RTS		RETURN

* CLOSE MACRO SPACE

0EF8 7D 00 92 CLSMAC TST DIVFL2 TEST MARKER
 0EFB 27 09 BEQ CLSMAC2

0EFD 4F		CLR R		
0EFE 97 91		STA A	DIVFLG	CLEAR FLAG
0F00 97 92		STA A	DIVFL2	
0F02 DE AE		LDX	NXTOUT	SET POINTER
0F04 20 02		BRA	CLSMR3	
0F06 DE AC	CLSMR2	LDX	NXTMAC	POINT TO NEXT MAC
0F08 6D 00	CLSMR3	TST	0, X	TEST CHARACTER
0F0A 27 14		BEQ	CLSMR4	
0F0C 6D 01		TST	1, X	TEST NEXT
0F0E 27 17		BEQ	CLSMR5	
0F10 6D 02		TST	2, X	ONE MORE
0F12 27 1A		BEQ	CLSMR6	
0F14 6F 00		CLR	B, X	CLEAR OUT SPACE
0F16 6F 01		CLR	1, X	
0F18 6F 02		CLR	2, X	
0F1A 08		INX		FIX POINTER
0F1B 08		INX		
0F1C 08		INX		
0F1D DF A6		STX	FSTAVL	SET FIRST AVAIL
0F1F 39		RTS		RETURN
0F20 EE 01	CLSMR4	LDX	1, X	GET LINK
0F22 26 E4		BNE	CLSMR3	
0F24 7E 0E E0		JMP	SYSERR	REPORT MACRO ERROR
0F27 86 FF	CLSMR5	LDA A	#\$FF	SET UP FF
0F29 A7 00		STA A	0, X	SAVE IT
0F2B 08		INX		
0F2C 20 F2		BRA	CLSMR4	
0F2E 86 FF	CLSMR6	LDA A	#\$FF	SET UP FF
0F30 A7 00		STA A	0, X	SAVE IT
0F32 08		INX		FIX POINTER
0F33 20 F2		BRA	CLSMR5	

* FIND MACRO

0F35 96 A0	FNDMAC	LDA A	MACNAM	CHECK NAME
0F37 27 17		BEQ	FNDMA4	
0F39 D6 A1		LDA B	MACNAM+1	GET NAME
0F3B CE 19 72		LDX	#MACTBL	POINT TO TABLE
0F3E 9C FA	FNDMA1	CPX	MACEND	FINISHED?
0F40 27 0E		BEQ	FNDMA4	
0F42 A1 00		CMP A	0, X	TEST 1ST CHAR.
0F44 26 04		BNE	FNDMA2	
0F46 E1 01		CMP B	1, X	TEST 2ND CHAR.
0F48 27 08		BEQ	FNDMA6	
0F4A 08	FNDMA2	INX		FIX POINTER
0F4B 08		INX		
0F4C 08		INX		
0F4D 08		INX		
0F4E 20 EE		BRA	FNDMA1	REPEAT
0F50 DE FA	FNDMA4	LDX	MACEND	SET POINTER
0F52 39	FNDMA6	RTS		RETURN

* FIND LAST MACRO ENTRY

0F53 A6 00	FNDLST	LDR A	0,X	GET CHARACTER
0F55 27 03		BEQ	FNDLS2	IS IT ZERO?
0F57 08		INX		GO TO NEXT
0F58 20 F9		BRA	FNDLST	
0F5A 08	FNDLS2	INX		BUMP POINTER
0F5B DF AC		STX	NXTMAC	SAVE POSITION
0F5D EE 00		LDX	0,X	PICK UP LINK
0F5F 26 F2		BNE	FNDLST	
0F61 DE AC		LDX	NXTMAC	GET NEXT LOC.
0F63 39		RTS		RETURN

* CHECK LAST MACRO ENTRY

0F64 8D ED	CHKLST	BSR	FNDLST	FIND LAST ENTRY
0F66 08		INX		FIX POINTER
0F67 08		INX		
0F68 9C A6		CPX	FSTRVL	IS IT FIRST?
0F6A 27 04		BEQ	CHKLS2	
0F6C DE AC		LDX	NXTMAC	GET NEXT
0F6E 0D		SEC		
0F6F 39		RTS		RETURN
0F70 09	CHKLS2	DEX		BACK UP
0F71 09		DEX		
0F72 09		DEX		
0F73 86 FF		LDA A	#\$FF	SET UP FF
0F75 A7 00		STA A	0,X	PUT CHARACTER
0F77 A7 01		STA A	1,X	
0F79 A7 02		STA A	2,X	
0F7B 0C		CLC		
0F7C 39		RTS		RETURN

* END MACRO EXECUTION

0F7D 96 8B	MCEND	LDA A	MACCNT	GET COUNT
0F7F 27 06		BEQ	MCEND2	
0F81 7F 00 66		CLR	MBFLG	CLEAR FLAG
0F84 7C 00 88		INC	FINMAC	SET FINISHED
0F87 7E 08 57	MCEND2	JMP	FINCM	GO FINISH

* GET TWO CHARACTER NAME

0F8A BD 11 D8	GTNAM	JSR	LDNSKP	GET TO NEXT
0F8D BD 11 E6		JSR	CLSFY	CLASSIFY IT
0F90 C1 02		CMP B	#2	
0F92 26 1C		BNE	GTNA6	
0F94 36		PSH A		SAVE CHARACTER
0F95 08		INX		FIX THE POINTER
0F96 A6 00		LDA A	0,X	GET CHARACTER
0F98 BD 11 E6		JSR	CLSFY	GO CLASSIFY
0F9B C1 02		CMP B	#2	
0F9D 33		PUL B		RESTORE CHARACTER
0F9E 26 10		BNE	GTNA6	
0FA0 08		INX		ADJUST POINTER
0FA1 DF E6		STX	CMNPNT	SAVE IT

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 42

0FA3 C1 5F		CMP B	#\$5F	LOWER CASE?
0FA5 23 04		BLS	GTNA4	
0FA7 80 20		SUB A	#\$20	MAKE UPPER
0FA9 C8 20		SUB B	#\$20	
0FAB D7 A0	GTNA4	STA B	MACNAM	SAVE THE NAME
0FAD 97 A1		STA A	MACNAM+1	
0FAF 39		RTS		RETURN
0FB0 4F	GTNA6	CLR A		CLEAR OUT
0FB1 5F		CLR B		
0FB2 20 F7		BRA	GTNA4	

* SET TITLE LENGTH . LT +N

0FB4 BD 12 1A	TLEN	JSR	CHKNUM	CHECK FOR NUMBER
0FB7 24 07		BCC	TLEN2	
0FB9 96 D8		LDA A	TLN	GET LENGTH
0FBB BD 12 09		JSR	FIXVAL	GO FIX VALUE
0FBE 97 D8		STA A	TLN	SAVE NEW
0FC0 39	TLEN2	RTS		RETURN

* DO THREE PART TITLE . TL '1'2'3'

0FC1 7F 00 B9	TITLE	CLR	TPOS	CLEAR POSITION
0FC4 BD 11 D8		JSR	LDNSKP	GET TO NEXT
0FC7 81 0D		CMP A	#\$D	C. R. ?
0FC9 27 F5		BEQ	TLEN2	
0FCB CE 18 BE		LDX	#TTLBUF	POINT TO BUFFER
0FCE 7C 00 69		INC	NOCR	SET FLAG
0FD1 DF BD	TITLE1	STX	TTLPNT	SAVE POINTER
0FD3 BD 96 8C		JSR	GETCHR	GO GET CHAR.
0FD6 DE BD		LDX	TTLPNT	RESTORE POINTER
0FD8 A7 00		STA A	0,X	SAVE THE CHAR.
0FDA 81 0D		CMP A	#\$D	FINISHED?
0FDC 27 03		BEQ	TITL12	
0FDE 08		INX		BUMP THE POINTER
0FDF 20 F0		BRA	TITLE1	
0FE1 7F 00 69	TITL12	CLR	NOCR	CLEAR FLAG
0FE4 CE 18 0R		LDX	#CMNDBF	POINT TO BUFFER
0FE7 A6 84	TITL15	LDA A	TTLBUF-CMNDBF,X	
0FE9 A7 00		STA A	0/X	PUT CHAR.
0FEB 08		INX		GET TO NEXT
0FEC 81 0D		CMP A	#\$D	FINISHED?
0FEE 26 F7		BNE	TITL15	
0FF0 CE 18 0A		LDX	#CMNDBF	RESTORE POINTER
0FF3 A6 00		LDA A	0,X	GET CHARACTER
0FF5 97 0A		STA A	DELIM	SAVE DELIMITER
0FF7 08		INX		BUMP THE POINTER
0FF8 DF E6		STX	CMNPNT	SAVE IT
0FFA CE 18 BE		LDX	#TTLBUF	POINT TO BUFFER
0FFD DF BD		STX	TTLPNT	
0FFF 86 20		LDA A	#\$20	SET UP SPACE
1001 A7 00	TITLE2	STA A	0,X	SAVE IT
1003 08		INX		BUMP POINTER
1004 8C 19 0E		CPY	#TTLBUF+80	

1007 26 F8	BNE	TITLE2	
1009 BD 10 72	JSR	CNTTTL	GO COUNT TITLE
100C D7 BB	STA B	TCNT	SAVE COUNT
100E BD 10 86	JSR	XFRRTL	TRANSFER TITLE
1011 BD 10 72	JSR	CNTTTL	COUNT TITLE
1014 96 D8	LDA A	TLN	GET LENGTH
1016 10	SBR		
1017 47	ASR A		
1018 97 BC	STA A	MCNT	SAVE MIDDLE COUNT
101A C6 20	LDA B	#\$20	GET SPACE
101C 91 BB	CMP A	TCNT	
101E 23 0F	BLS	TITLE5	
1020 90 BB	SUB A	TCNT	
1022 DE BD	LDX	TTLPNT	RESTORE POINTER
1024 E7 00	STA B	0, X	SAVE CHAR.
1026 08	INX		BUMP THE POINTER
1027 7C 00 B9	INC	TPOS	UPDATE POSITION
102A 4A	DEC A		
102B 26 F7	BNE	TITLE4	
102D DF BD	STX	TTLPNT	SAVE POINTER
102F BD 10 86	JSR	XFRRTL	TRANSFER TITLE
1032 BD 10 72	JSR	CNTTTL	COUNT TITLE
1035 96 D8	LDA A	TLN	GET LENGTH
1037 90 B9	SUB A	TPOS	FIX POSITION
1039 11	CBA		
103A 23 0D	BLS	TITLE7	
103C 10	SBR		
103D C6 20	LDA B	#\$20	SET UP SPACE
103F DE BD	LDX	TTLPNT	SET POINTER
1041 E7 00	STA B	0, X	PUT CHAR
1043 08	INX		BUMP POINTER
1044 4A	DEC A		DEC THE COUNT
1045 26 FA	BNE	TITL65	
1047 DF BD	STX	TTLPNT	SAVE POINTER
1049 BD 10 86	JSR	XFRRTL	TRANSFER TITLE
104C 96 D8	LDA A	TLN	GET LENGTH
104E 97 B9	STA A	TPOS	SAVE POSITION
1050 27 1C	BEQ	TITLE9	
1052 D6 4E	LDA B	LFM	CHECK MARGIN
1054 27 0A	BEQ	TITL78	
1056 86 20	STA A	#\$20	SETUP SPACE
1058 37	PSH B		
1059 BD 14 9F	JSR	OUTCHR	OUTPUT SPACE
105C 33	PUL B		
105D 5A	DEC B		DEC COUNT
105E 26 F6	BNE	TITL75	
1060 CE 18 BE	LDX	#TTLBUF	POINT TO TITLE
1063 A6 00	LDA A	0, X	GET A CHARACTER
1065 BD 14 9F	JSR	OUTCHR	OUTPUT IT
1068 08	INX		GO TO NEXT
1069 7A 00 B9	DEC	TPOS	DEC COUNT
106C 26 F5	BNE	TITLE8	REPEAT TIL DONE
106E BD 08 99	JSR	PCRLF	OUTPUT CR & LF
1071 39	RTS		RETURN

* COUNT CHARACTERS IN TITLE

1072 5F	CNTTTL	CLR B		CLEAR COUNT
1073 DE E6		LDX CMNPNT		SET POINTER
1075 A6 00	CNTTT2	LDA A 0,X		GET CHARACTER
1077 91 BA		CMP A DELIM		IS IT DELIMITER?
1079 27 08		BEQ CNTTT3		
107B 81 0D		CMP A #\$D		IS IT C.R.?
107D 27 04		BEQ CNTTT3		
107F 08		INX		BUMP THE POINTER
1080 5C		INC B		BUMP COUNT
1081 20 F2		BRA CNTTT2		
1083 DF E8	CNTTT3	STX SPCPT1		SET POINTER
1085 39		RTS		RETURN

* TRANSFER TITLE TO BUFFER

1086 DE E6	XFR TTL	LDX CMNPNT		SET POINTER
1088 9C E8		CPX SPCPT1		FINISHED?
108A 27 15		BEQ BMPCP2		
108C A6 00		LDA A 0,X		GET CHARACTER
108E 08		INX		BUMP TO NEXT
108F DF E6		STX CMNPNT		SAVE
1091 DE BD		LDX TTL PNT		SET POINTER
1093 A7 00		STA A 0,X		PUT CHARACTER
1095 08		INX		BUMP TO NEXT
1096 DF BD		STX TTL PNT		SAVE
1098 7C 00 B9		INC TPOS		BUMP POSITION
109B 20 E9		BRA XFR TTL		REPEAT
109D 20 02	XFR TT2	BRA BMPCP2		

* BUMP COMMAND POINTER

109F DE E6	BMPCP	LDX CMNPNT		GET POINTER
10A1 08	BMPCP2	INX		BUMP IT
10A2 DF E6		STX CMNPNT		SAVE IT
10A4 39		RTS		RETURN

* IF COMMAND . IF CONDITION . CM

10B5 7F 00 89	IF	CLR NEGT		CLEAR FLAG
10B8 BD 11 D8	IF1	JSR LDNSKP		FIND NEXT CHAR
10B9 91 21		CMP A #'!'		IS IT A '!'?
10BD 26 07		BNE IF3		
10BF 73 00 89		COM NEGT		SET NEG FLAG
10B2 8D EB		BSR BMPCP		BUMP POINTER
10B4 20 F2		BRA IF1		
10B6 81 5F	IF3	CMP A #\$5F		IS IT LOWER CASE?
10B8 23 02		BLS IF35		
10B9 80 20		SUB A #\$20		MAKE UPPER
10BC 81 4F	IF35	CMP A #'0		CHECK IF ODD
10BE 26 07		BNE IF4		
10C0 96 74		LDA A PGN		GET PAGE NUMBER

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 45

10C2 46		ROR A	CHECK IF ODD	
10C3 24 28		BCC	IFN	
10C5 20 09		BRA	IFY	
10C7 81 45	IF4	CMP A	#'E	EVEN?
10C9 26 27		BNE	IF6	
10CB 96 74		LDA A	PGN	GET PAGE NUMBER
10CD 46		ROR A		CHECK IF EVEN
10CE 25 1D		BCS	IFN	
10D0 96 89	IFY	LDA A	NEGT	CHECK NEG.
10D2 26 31		BNE	IF8	
10D4 8D C9	IF5	BSR	BMPCP	BUMP POINTER
10D6 BD 11 D8		JSR	LDNSKP	GET NEXT CHAR
10D9 7F 00 6F		CLR	NOFL	CLEAR FLAG
10DC 81 2E		CMP A	#'	IS IT PERIOD?
10DE 27 06		BEQ	IF55	
10E0 81 3A		CMP A	#':	IS IT COLON?
10E2 26 00		BNE	IFN2	
10E4 97 6F		STA A	NOFL	SET NO FLUSH
10E6 08	IF55	INX		FIX POINTER
10E7 DF E6		STX	CMNPNT	SAVE IT
10E9 7C 00 8A		INC	IFFLG	SET IF FLAG
10EC 39		RTS		RETURN
10ED 96 89	IFN	LDA A	NEGT	CHECK NEG.
10EF 26 E3		BNE	IF5	
10F1 39		RTS		RETURN
10F2 BD 12 1A	IF6	JSR	CHKNUM	CHECK FOR NUMBER
10F5 24 0E		BCC	IF8	
10F7 DE E6		LDX	CMNPNT	GET POINTER
10F9 09		DEX		ADJUST
10FA 09		DEX		
10FB DF E6		STX	CMNPNT	SAVE
10FD 96 70		LDA A	INNUM	GET NUMBER
10FF 2B EC		BMI	IFN	
1101 27 EA		BEQ	IFN	
1103 20 CB		BRA	IFY	
1105 39	IF8	RTS		RETURN

* TEST FOR NEGATIVE NUMBER

1106 96 70	TSTNEG	LDA A	INNUM	GET NUMBER
1108 2A 07		BPL	TSTNE2	
110A 97 72		STA A	SIGN	SET SIGN
110C 97 71		STA A	NEG	SET NEG
110E 70 00 70		NEG	INNUM	NEGATE NUM.
1111 39	TSTNE2	RTS		RETURN

* CHANGE TRAP LOCATION CH -M -N

1112 BD 12 1A	CHNG	JSR	CHKNUM	CHECK FOR NUMBER
1115 24 19		BCC	CHNG3	
1117 BD ED		BSR	TSTNEG	NEGATIVE?
1119 96 4F		LDA A	PGL	GET PAGE LENGTH
111B 4C		INC A		
111C BD 12 09		JSR	FIXVAL	FIX VALUE

111F CE 17 D8		LDX	#TRAPS	POINT TO TRAPS
1122 4D		TST A		
1123 26 01		BNE	CHNG2	
1125 4C		INC A		BUMP IT
1126 A1 00	CHNG2	CMP A	0,X	TEST LOCATION
1128 27 23		BEQ	CHNG5	
112A BD 0D 97		JSR	INTRP	BUMP POS.
112D 26 F7		BNE	CHNG2	
112F 39	CHNG25	RTS		RETURN
1130 BD 0F 8A	CHNG3	JSR	GTNAM	GO GET NAME
1133 96 A8		LDA A	MACNAM	
1135 27 F8		BEQ	CHNG25	
1137 D6 A1		LDA B	MACNAM+1	
1139 CE 17 D8		LDX	#TRAPS	POINT TO TRAPS
113C A1 01	CHNG4	CMP A	1,X	CHECK CHAR.
113E 26 04		BNE	CHNG45	
1140 E1 82		CMP B	2,X	
1142 27 09		BEQ	CHNG5	
1144 08	CHNG45	INX		BUMP TO NEXT
1145 08		INX		
1146 08		INX		
1147 8C 18 08		CPX	#TRPEND	END OF TABLE?
114A 26 F0		BNE	CHNG4	
114C 39		RTS		RETURN
114D DF F0	CHNG5	STX	TEMPS	SAVE POINTER
114F BD 12 1A		JSR	CHKNUM	CHECK FOR NUMBER
1152 24 DB		BCC	CHNG25	
1154 BD B0		BSR	TSTNEG	IS IT NEG. ?
1156 96 4F		LDA R	PGL	GET PAGE LENGTH
1158 4C		INC A		
1159 BD 12 09		JSR	FIXVAL	FIX VALUE
115C 4D		TST A		
115D 26 01		BNE	CHNG6	
115F 4C		INC A		BUMP IT
1160 DE F0	CHNG6	LDX	TEMPS	RESTORE POINTER
1162 A7 00		STA R	0,X	PUT CHAR
1164 39		RTS		RETURN

* SET NUMBER REGISTER . NR X N

1165 BD 11 D8	NREG	JSR	LDNSKP	GET TO NEXT
1168 BD 11 E6		JSR	CLSFY	CLASSIFY IT
116B C1 02		CMP B	#2	
116D 26 14		BNE	NREG4	
116F 36		PSH A		SAVE
1170 BD 10 9F		JSR	BMPCP	BUMP POINTER
1173 BD 12 1A		JSR	CHKNUM	CHECK FOR NUMBER
1176 32		PUL A		RESTORE
1177 24 0A		BCC	NREG4	
1179 BD 11 A8		JSR	FNDNUM	GO FIND NUMBER
117C A6 00		LDA A	0,X	GET CHARACTER
117E BD 12 09		JSR	FIXVAL	FIX VALUE
1181 A7 00		STA A	0,X	SAVE IT
1183 39	NREG4	RTS		RETURN

* SET ARABIC MODE .AR

1184 7F 00 C4	ARB	CLR	ROM	CLEAR ROMAN
1187 39		RTS		RETURN

* SET FOR SMALL ROMAN .SR

1188 86 80	SROM	LDA A	##\$80	
118A 97 C4	ROM2	STA A	ROM	SET FLAG
118C 39		RTS		

* SET FOR CAPITAL ROMAN .CR

118D 86 0F	CROM	LDA A	##\$F	
118F 20 F9		BRA	ROM2	SET FLAG

* SET AUTO INCREMENT .AU N

1191 BD 12 1A	SAUTO	JSR	CHKNUM	CHECK FOR NUMBER
1194 24 07		BCC	SAUTO4	
1196 96 C2		LDA A	AUTO	GET OLD
1198 BD 12 09		JSR	FIXVAL	FIX VALUE
119B 97 C2		STA A	AUTO	SAVE NEW
119D 39	SAUTO4	RTS		RETURN

* CLEAR NUMBER SPACE

119E 5F	CLRNUM	CLR B		
119F D7 70		STA B	INNUM	CLEAR OUT NUM
11A1 D7 5F		STA B	INC	
11A3 D7 5A		STA B	GNUM	SET FLAGS
11A5 D7 5D		STA B	BNUM	
11A7 39		RTS		RETURN

* FIND NUMBER REGISTER

11A8 CE 00 40	FNDNUM	LDX	#NMREGS	SET POINTER
11AB 80 41		SUB R	##\$41	
11AD DF 60		STX	NUMPNT	
11AF 9B 61		ADD R	NUMPNT+1	ADD OFFSET
11B1 97 61		STA A	NUMPNT+1	
11B3 DE 60		LDX	NUMPNT	GET POINTER
11B5 39		RTS		RETURN

* FETCH NUMBER FROM BUFFER

11B6 DE 60	FTCHNM	LDX	NUMPNT	SET POINTER
11B8 9C 63		CPX	LSTNUM	FINISHED?
11BA 27 16		BEQ	FTCHN2	
11BC A6 00		LDA A	0,X	GET A CHAR.
11BE 84 7F		AND R	##\$7F	MASK IT
11C0 08		INX		BUMP THE POINTER
11C1 DF 60		STX	NUMPNT	SAVE IT

11C3 81 0D	CMP A	#\$D	C. R. ?
11C5 26 08	BNE	FTCHN1	
11C7 7D 00 69	TST	NOCR	TEST FLAG
11CA 26 03	BNE	FTCHN1	
11CC 7E 07 48	JMP	FETCH5	RETURN
11CF 7E 07 0A	FTCHN1	JMP	FETCH3
11D2 7F 00 62	FTCHN2	CLR	EXCHR
11D5 7E 06 8C		JMP	GETCHR
			CLEAR EXTRA CHAR.
			GO GET CHAR

* LOAD POINTER AND SKIP SPACES

11D8 DE E6	LDNSKP	LDX	CMNPNT	SET POINTER
11DA A6 00	LDNSK2	LDA A	0,X	GET CHARACTER
11DC 81 20		CMP A	#\$20	IS IT SPACE?
11DE 26 03		BNE	LDNSK4	
11E0 08		INX		BUMP TO NEXT
11E1 20 F7		BRA	LDNSK2	
11E3 DF E6	LDNSK4	STX	CMNPNT	SAVE POSITION
11E5 39		RTS		RETURN

* CLASSIFY CHARACTER

11E6 5F	CLSFY	CLR B		CLEAR SPECIFIER
11E7 4D		TST A		TEST CHAR
11E8 2B 1E		BMI	CLSFY4	
11EA 81 5F		CMP A	#\$5F	LOWER CASE?
11EC 23 06		BLS	CLSFY1	
11EE 81 7F		CMP A	#\$7F	TEST FOR PARITY
11F0 22 16		BHI	CLSFY4	
11F2 80 20		SUB A	#\$20	MAKE UPPER CASE
11F4 81 30	CLSFY1	CMP A	#\$'0	CHAR A NUMBER?
11F6 25 10		BLO	CLSFY4	
11F8 81 39		CMP A	#\$'9	
11FA 22 02		BHI	CLSFY2	
11FC 5C		INC B		IF SO, SET
11FD 39		RTS		RETURN
11FE 81 41	CLSFY2	CMP A	#\$'A	IS CHAR A LETTER?
1200 25 06		BLO	CLSFY4	
1202 81 5A		CMP A	#\$'Z	
1204 22 02		BHI	CLSFY4	
1206 C6 02		LDA B	#\$2	IF SO, SET
1208 39	CLSFY4	RTS		RETURN

* FIX NUMBER VALUE

1209 D6 70	FIXVAL	LDA B	INNUM	GET NUMBER
120B 7D 00 72		TST	SIGN	TEST SIGN
120E 27 08		BEQ	FIXVA4	
1210 7D 00 71		TST	NEG	TEST FOR NEG.
1213 27 81		BEQ	FIXVA3	
1215 50		NEG B		NEGATE NUM
1216 18	FIXVA3	ABA		FIX VALUE
1217 39		RTS		RETURN
1218 17	FIXVA4	TBA		

1219 39

RTS

* CHECK FOR NUMBER

1218 4F	CHKNUM	CLR A	CLEAR FLAGS	
121B 97 72		STA A	SIGN	
121D 97 71		STA A	NEG	
121F BD 11 9E		JSR	CLRNUM	CLEAR NUMBER
1222 5C		INC B		
1223 D7 69		STA B	NOCR	SET FLAGS
1225 BD 11 D8		JSR	LDNSKP	GO TO NEXT
1228 81 28		CMP A	#'+	IS IT A '+'?
122A 27 06		BEQ	CHKNU2	
122C 81 2D		CMP A	#'-	IS IT A '-'?
122E 26 0B		BNE	CHKNU4	
1230 97 71		STA A	NEG	SET NEG.
1232 08	CHKNU2	INX	BUMP THE POINTER	
1233 DF E6		STX	CMNPNT	SAVE IT
1235 8D 1F		BSR	PRNUM	PROCESS NUMBER
1237 24 0D		BCC	CHKNU6	
1239 20 07		BRA	CHKNU5	
123B 08	CHKNU4	INX	FIX POINTER	
123C DF E6		STX	CMNPNT	SAVE IT
123E 8D 32		BSR	PRNU28	PROCESS NUM.
1240 24 04		BCC	CHKNU6	
1242 8D 0B	CHKNU5	BSR	CLRTHM	CLEAR FLAGS
1244 0D		SEC		
1245 39		RTS	RETURN	
1246 8D 07	CHKNU6	BSR	CLRTHM	CLEAR FLAGS
1248 DE E6		LDX	CMNPNT	SET POINTER
124A 09		DEX		
124B DF E6		STX	CMNPNT	
124D 0C		CLC		
124E 39		RTS	RETURN	

* CLEAR FLAGS

124F 7F 00 62	CLRTHM	CLR	EXCHR	CLEAR THEM
1252 7F 00 69		CLR	NOCR	
1255 39		RTS	RETURN	

* PROCESS NUMBER

1256 97 72	PRNUM	STA A	SIGN	CLEAR SIGN.
1258 BD 11 9E	PRNUM2	JSR	CLRNUM	CLEAR NUMBER
125B 5C		INC B		
125C D7 69		STA B	NOCR	SET FLAGS
125E 7F 00 5F	PRNU27	CLR	INC	
1261 7C 00 75		INC	PASCHR	
1264 BD 06 8C		JSR	GETCHR	GET NEXT CHAR.
1267 7D 00 7F		TST	NOEXP	DO EXPRESSIONS?
126A 27 06		BEQ	PRNU28	
126C 7F 00 7F		CLR	NOEXP	
126F 7E 13 19		JMP	PRNU82	JUMP AHEAD

1272 BD 11 E6	PRNU28	JSR	CLSFY	GO CLASSIFY
1275 C1 01		CMP B	#1	
1277 25 05	PRNUM3	BLO	PRNU31	
1279 27 51		BEQ	PRNUM5	
127B 7E 13 07		JMP	PRNU73	
127E 7D 00 70	PRNU31	TST	INNUM	TEST NUMBER
1281 27 05		BEQ	PRNU32	
1283 36		PSH A		
1284 96 70		LDA A	INNUM	GET NUMBER
1286 20 58		BRA	PRNUM6	
1288 7F 00 70	PRNU32	CLR	INNUM	CLEAR NUMBER
128B 81 23		CMP A	#'#	CHECK FOR '#'
128D 27 18		BEQ	PRNUM4	
128F 81 2B		CMP A	#'+	IS IT '+/?
1291 26 04		BNE	PRNU35	
1293 97 5B		STA A	ADD	SET FOR ADD
1295 20 C7		BRA	PRNU27	
1297 81 2D	PRNU35	CMP A	#'-	IS IT '-/?
1299 26 04		BNE	PRNU37	
129B 97 5C		STA A	SUB	SET FOR SUBTRACT
129D 20 BF		BRA	PRNU27	
129F 81 25	PRNU37	CMP A	#'%	IS IT '%/?
12A1 26 6C		BNE	PRNUM8	
12A3 96 74		LDA A	PGN	GET PAGE NUMBER
12A5 20 39		BRA	PRNUM6	
12A7 7C 00 75	PRNUM4	INC	PASCHR	SET FLAG
12AA BD 06 8C		JSR	GETCHR	GET CHARACTER
12AD BD 11 E6		JSR	CLSFY	CLASSIFY IT
12B0 C1 02		CMP B	#2	
12B2 26 10		BNE	PRNU45	
12B4 BD 11 A8		JSR	FNDNUM	GO FIND NUMBER
12B7 A6 00		LDA A	0, X	GET VALUE
12B9 7D 00 5F		TST	INC	INCREMENT?
12BC 27 22		BEQ	PRNUM6	
12BE 9B C2		ADD A	AUTO	ADD IN AUTO
12C0 A7 00		STA A	0, X	SAVE NEW
12C2 20 1C		BRA	PRNUM6	
12C4 81 2B	PRNU45	CMP A	#'+	IS IT '+/?
12C6 26 47		BNE	PRNUM8	
12C8 97 5F		STA A	INC	SET INC.
12CA 20 DB		BRA	PRNUM4	
12CC 80 30	PRNUM5	SUB A	#\$30	BIAS NUMBER
12CE 36		PSH A		
12CF D6 70		LDA B	INNUM	GET NUM
12D1 58		ASL B		ADJUST
12D2 58		ASL B		
12D3 DB 70		ADD B	INNUM	ADD IT IN
12D5 58		ASL B		
12D6 32		PUL A		RESTORE
12D7 1B		ABA		
12D8 97 70		STA A	INNUM	SAVE NEW VALUE
12DA 7C 00 5A		INC	GNDNUM	SET GOOD
12DD 7E 12 5E		JMP	PRNU27	REPEAT
12E0 D6 5C	PRNUM6	LDA B	SUB	SUBTRACT?

12E2 27 06		BEQ	PRNU65	
12E4 16		TAB		DO SUBTRACT
12E5 96 5D		LDA A	BNUM	
12E7 10		SBA		
12E8 20 06		BRA	PRNUM7	
12EA D6 5B	PRNU65	LDA B	ADD	ADDITION?
12EC 27 02		BEQ	PRNUM7	
12EE 9B 5D		ADD A	BNUM	DO ADD
12F0 97 5D	PRNUM7	STA A	BNUM	SAVE NUMBER
12F2 7F 00 5B		CLR	ADD	CLEAR FLAGS
12F3 7F 00 5C		CLR	SUB	
12F8 7C 00 5A		INC	GNDUM	SET GOOD
12FB 7D 00 70		TST	INNUM	TEST NUMBER
12FE 26 03		BNE	PRNU72	
1300 7E 12 5E		JMP	PRNU27	
1303 32	PRNU72	PUL A		RESTORE CHAR
1304 7E 12 88		JMP	PRNU32	
1307 7D 00 70	PRNU73	TST	INNUM	TEST NUMBER
130A 27 03		BEQ	PRNUM8	
130C 36		PSH A		
130D 20 D1		BRA	PRNUM6	
130F 7F 00 7F	PRNUM8	CLR	NOEXP	CLEAR FLAG
1312 7D 00 5A		TST	GNDUM	TEST GOOD
1315 26 02		BNE	PRNU82	
1317 0C		CLC		SET CONDITION
1318 39		RTS		RETURN
1319 97 62	PRNU82	STA A	EXCHR	SAVE EXTRA CHAR.
131B CE 01 25		LDX	#NUM	POINT TO NUMBER
131E 96 5D		LDA A	BNUM	GET NUMBER
1320 97 70		STA A	INNUM	
1322 27 04		BEQ	BTOD	
1324 D6 C4		LDA B	ROM	ROMAN OR ARABIC?
1326 26 37		BNE	BTOROM	

* BINARY TO ASCII ARABIC

1328 5F		BTOD	CLR B	
1329 81 64		BTOD1	CMP A #100	NUM > 100?
132B 25 05			BLO BTOD2	
132D 80 64			SUB A #100	SUB OFF 100
132F 5C			INC B	BUMP NUMBER
1330 20 F7			BRA BTOD1	
1332 5D	BTOD2	TST B		ANY YET?
1333 27 06		BEQ	BTOD3	
1335 CB 30		ADD B #\$30		SET HUNDREDS
1337 E7 00		STA B 0,X		SAVE
1339 08		INX		GO TO NEXT
133A 5F		CLR B		CLEAR REGISTER
133B 81 0A	BTOD3	CMP A #10		NUMBER > 10
133D 25 05		BLO BTOD4		
133F 80 0A		SUB A #10		SUB VALUE
1341 5C		INC B		BUMP NUMBER
1342 20 F7		BRA BTOD3		
1344 5D	BTOD4	TST B		ANY?

1345 C7 05		BEQ	BTOD45	
1347 CB 30		ADD B	#\$30	ADD BIAS
1349 E7 00		STA D	0,X	SAVE TENS
134B 08		INX		BUMP TO NEXT
134C 88 30	BTOD45	ADD A	#\$30	ADD IN BIAS
134E A7 00		STA H	0,X	SAVE ONES
1350 08		INX		BUMP POINTER
1351 96 62	BTOD5	LDA H	EXCHR	GET EXTRA
1353 A7 00		STA A	0,X	SAVE IT
1355 08		INX		BUMP TO NEXT
1356 DF 63		STX	LSTNUM	SAVE POSITION
1358 CE 01 25		LDX	#NUM	POINT TO NUMBER
135B DF 60		STX	NUMPNT	
135D 0D		SEC		
135E 39		RTS		RETURN

* BINARY TO ASCII ROMAN

135F C6 43		BTOROM	LDA B	'C	SET HUNDREDS
1361 81 64		BTOR01	CMP R	#100	NUMBER > 100?
1363 25 07		BLO	BTOR02		
1365 80 64		SUB R	#100	SUBTRACT OFF	
1367 E7 00		STA B	0,X	SET 100	
1369 08		INX		BUMP TO NEXT	
136A 20 F5		BRA	BTOR01		
136C 81 5A	BTOR02	CMP R	#90	CHECK FOR 90	
136E 25 0A		BLO	BTOR03		
1370 80 5A		SUB R	#90	SUBTRACT OFF	
1372 E7 01		STA B	1,X	PUT CHARACTER	
1374 C6 58		LDA B	'X	SET TENS	
1376 E7 00		STA B	0,X	SAVE IT	
1378 08		INX		BUMP TO NEXT	
1379 08		INX			
137A 81 32	BTOR03	CMP H	#50	CHECK FOR FIFTY	
137C 25 07		BLO	BTOR04		
137E 80 32		SUB H	#50	SUBTRACT OFF	
1380 C6 40		LDA H	'L	SET 'L'	
1382 E7 00		STA H	0,X	SAVE IT	
1384 08		INX		BUMP THE POINTER	
1385 81 28	BTOR04	CMP H	#40	CHECK FOR 40	
1387 25 0C		BLO	BTOR05		
1389 80 28		SUB H	#40	SUBTRACT OFF	
138B C6 58		LDA H	'X	SET TEN	
138D E7 00		STA H	0,X	SAVE IT	
138F C6 4C		LDA H	'L	SET 50	
1391 E7 01		STA H	1,X	SAVE IT	
1393 08		INX		BUMP TO NEXT	
1394 08		INX			
1395 C6 58	BTOR05	LDA H	'X'	SET UP 'X'	
1397 81 0A		CMP H	#10	CHECK TENS	
1399 25 07		BLO	BTOR06		
139B 80 0A		SUB H	#10	SUBTRACT OFF	
139D E7 00		STA H	0,X	SAVE	
139F 08		INX		BUMP POINTER	

13A0 20 F3		BRA	BT0R05	
13A2 81 09		BTO	CMP A #9	CHECK IF 9
13A4 25 0A		BLO	BT0R65	
13A6 80 09		SUB A	#9	SUBTRACT 9
13A8 E7 01		STA B	1,X	SAVE CHARACTER
13AA C6 49		LDA B	#'I	
13AC E7 00		STA B	0,X	
13AE 08		INX		GET TO NEXT
13AF 08		INX		
13B0 81 05	BT0R06	CMP A	#5	CHECK FOR 5
13B2 25 07		BLO	BT0R07	
13B4 C6 56		LDA B	#'V	SET UP 'V'
13B6 E7 00		STA B	0,X	SAVE IT
13B8 08		INX		BUMP POINTER
13B9 80 05		SUB A	#5	FIX VALUE
13BB 81 04	BT0R07	CMP A	#4	CHECK FOR 4
13BD 25 0C		BLO	BT0R08	
13BF 80 04		SUB A	#4	SUBTRACT OFF
13C1 C6 49		LDA B	#'I	SET UP 'I'
13C3 E7 00		STA B	0,X	SAVE CHARACTER
13C5 C6 56		LDA B	#'V	
13C7 E7 01		STA B	1,X	SAVE 'V'
13C9 08		INX		BUMP POINTER
13CA 08		INX		
13CB C6 49	BT0R08	LDA B	#'I	
13CD 4D		TST A		TEST ONES
13CE 27 06		BEQ	BT0R09	
13D0 E7 00		STA B	0,X	SAVE I'S
13D2 08		INX		
13D3 4A		DEC A		DONE?
13D4 20 F5		BRA	BT0R08	
13D6 DF 63	BT0R09	STX	LSTNUM	SAVE POINTER
13D8 96 C4		LDA A	ROM	CHECK IF SMALL
13DA 2A 0E		BPL	BTODON	
13DC CE 01 25		LDX	#NUM	RESET POINTER
13DF A6 00	BT0R92	LDA A	0,X	GET CHARACTER
13E1 8B 20		ADD A	#\$20	MAKE SMALL
13E3 A7 00		STA A	0,X	PUT BACK
13E5 08		INX		BUMP TO NEXT
13E6 9C 63		CPX	LSTNUM	FINISHED?
13E8 26 F5		BNE	BT0R92	
13EA 7E 13 51	BTODON	JMP	BT0D5	

* PUSH X ONTO STACK

13ED 32	PUSHX	PUL A		GET RETURN ADR.
13EE 33		PUL B		
13EF 97 F4		STA A	RETREG	SAVE IT
13F1 D7 F5		STA B	RETREG+1	
13F3 DF F6		STX	INDEX	SAVE X
13F5 96 F6		LDA A	INDEX	GET PART X
13F7 D6 F7		LDA B	INDEX+1	
13F9 36		PSH A		PUSH ON STACK
13FA 37		PSH B		

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 54

13FB DE F4	PUSH4	LDX	RETREG	GET RETURN
13FD 6E 00		JMP	0,X	RETURN

* PULL X FROM STACK

13FF 32	PULLX	PUL A	GET RETURN ADR	
1400 33		-PUL B		
1401 97 F4		STA A	RETREG	SAVE IT
1403 D7 F5		STA B	RETREG+1	
1405 33		PUL B		FULL X
1406 32		PUL A		
1407 97 F6		STA A	INDEX	SAVE X
1409 D7 F7		STA B	INDEX+1	
140B 96 F4		LDA A	RETREG	GET RETURN ADR.
140D D6 F5		LDA B	RETREG+1	
140F 37		PSH B		PUSH BACK ON
1410 36		PSH A		
1411 DE F6		LDX	INDEX	LOAD UP X
1413 39		RTS		RETURN

* FIX WIDTH

1414 96 B4	FIXWD	LDA A	TLLN	GET TEMP LENGTH
1416 9B 4B		ADD A	LLN	ADD TO LENGTH
1418 97 4B		STA A	LLN	SAVE NEW
141A 96 B3		LDA A	TIND	GET TEMP IND.
141C 9B 48		ADD A	IND	ADD TO INDENT
141E 97 48		STA A	IND	SAVE NEW
1420 96 B2		LDA A	TSIN	GET TEMP SIND.
1422 9B 7C		ADD A	SIN	ADD TO SIND.
1424 97 7C		STA A	SIN	SAVE NEW
1426 4F		CLR A		CLEAR OLD VALUES
1427 97 B4		STA A	TLLN	
1429 97 B3		STA A	TIND	
142B 97 B2		STA A	TSIN	
142D 96 4B		LDA A	LLN	GET LINE LENGTH
142F 90 48		SUB A	IND	SUB INDENT
1431 90 7C		SUB A	SIN	SUB S IND.
1433 81 0E		CMP A	#14	LESS THAN 15?
1435 22 02		BHI	FIXWD2	
1437 86 0F	FIXWD1	LDR A	#15	FORCE TO 15
1439 81 96	FIXWD2	CMP A	#150	>150?
143B 23 02		BLS	FIXWD3	
143D 86 96		LDA A	#150	
143F 97 C6	FIXWD3	STA A	WIDTH	SAVE NEW WIDTH
1441 7E 06 46		JMP	FIXBFE	GO FIX

* GET INPUT CHARACTERS

1444 CE 17 74	GIBUF	LDX	#SBUF	POINT TO BUFFER
1447 5F		CLR B		CLEAR COUNT
1448 37	GIBUF2	PSH B		
1449 BD 02 06		JSR	INCH	GET CHARACTER
144C 33		PUL B		

144D 81 18	CMP A	#\$18	CONTROL X?
144F 27 18	BEQ	GIBUF6	
1451 81 0D	CMP A	#\$D	C. R. ?
1453 27 0A	BEQ	GIBUF4	
1455 81 1F	CMP A	#\$1F	CONTROL CHAR. ?
1457 23 EF	BLS	GIBUF2	
1459 5C	INC B		BUMP THE COUNT
145A R7 08	STA A	0,X	PUT CHARACTER
145C 08	INX		BUMP THE POINTER
145D 20 E9	BRA	GIBUF2	REPEAT
145F A7 00	STA A	0,X	PUT CHARACTER
1461 CE 17 74	LDX	#\$BUF	FIX POINTER
1464 DF E6	STX	CMNPNT	SAVE IT
1466 D7 46	STA B	GCNT	SAVE COUNT
1468 39	RTS		RETURN
1469 CE 15 62	LDX	#QUSTR	POINT TO STRING
146C BD 14 90	JSR	PSTRNG	OUTPUT IT
146F 20 D3	BRA	GIBUF	

* TEST FOR BREAK

1471 B6 80 04	TSTBRK	LDA A	\$8004	GET STATUS
1474 2A 01		BPL	TSTBR2	
1476 39		RTS		RETURN
1477 B6 80 04	TSTBR2	LDA A	\$8004	GET STATUS
147A 2A FB		BPL	TSTBR2	WAIT TIL CLEAR
147C 01		NOP		FOR EXPANSION
147D 01		NOP		
147E 01		NOP		
147F CE 15 92		LDX	#BRKSTR	POINT TO STRING
1482 7E 0C D8		JMP	STOP1	OUTPUT IT

* OUTPUT A C. R. AND L. F.

1485 DF F8	CRLF	STX	XTEMP	SAVE X REG.
1487 CE 15 43		LDX	#CRLFST	POINT TO STRING
148A BD 14 92		JSR	PDATA	OUTPUT IT
148D DE F8		LDX	XTEMP	RESTORE X
148F 39	CRLFS	RTS		RETURN

* PRINT STRING

1490 8D F3	PSTRNG	BSR	CRLF	OUTPUT CR & LF
------------	--------	-----	------	----------------

* PRINT DATA

1492 R6 00	PDATA	LDA A	0,X	GET A CHARACTER
1494 81 04		CMP A	#4	IS IT TERM?
1496 27 06		BEQ	PDATA2	
1498 BD 02 03		JSR	OUTCH	OUTPUT IT
1498 08		INX		MOVE TO NEXT
149C 20 F4		BRA	PDATA	REPEAT
149E 39	PDATA2	RTS		RETURN

* OUTPUT CHARACTER

149F D6 91	OUTCHR	LDA B	DIVFLG	DIVERTING?
14A1 27 06		BEQ	DUTCH2	
14A3 7C 00 92		INC	DIVFL2	SET FLAG
14A6 7E 0E AE		JMP	OUTMAC	OUT TO MACRO
14A9 D6 79	DUTCH2	LDA B	NOOUT	DO OUTPUT?
14AB 27 01		BEQ	DUTCH3	
14AD 39		RTS		RETURN
14AE 84 7F	DUTCH3	AND A	#\$7F	MASK CHAR.
14B0 D6 93		LDA B	PRNTR	TO PRINTER?
14B2 26 03		BNE	DUTCH4	
14B4 7E 02 03		JMP	DUTCH	OUT TO TERM
14B7 7E 02 0F	DUTCH4	JMP	POUCH	OUT TO PRINTER

* INPUT A CHARACTER

14BA 96 FC	INCHR	LDA R	CRF	CHECK FLAG
14BC 27 0D		BEQ	INCH15	
14BE 37		PSH B		
14BF D6 9D		LDA B	JNKCNT	GET JUNK COUNT
14C1 27 05		BEQ	INCHR1	
14C3 8D 06	INCHR0	BSR	INCH15	SKIP JUNK
14C5 5A		DEC B		DEC COUNT
14C6 26 FB		BNE	INCHR0	
14C8 D7 FC	INCHR1	STA B	CRF	
14CA 33		PUL B		
14CB DF F8	INCH15	STX	XTEMP	SAVE X REG.
14CD DE 9B		LDX	NXTRAM	POINT TO NEXT
14CF 9C 99		CPX	LSTRAM	FINISHED?
14D1 27 0E		BEQ	INCHR4	
14D3 A6 00		LDA R	0,X	GET CHARACTER
14D5 08		INX		BUMP TO NEXT
14D6 DF 9B	INCHR2	STX	NXTRAM	SAVE POSITION
14D8 81 0D		CMP A	#\$D	IS IT C.R.?
14DA 26 02		BNE	INCHR3	
14DC 97 FC		STA A	CRF	SET FLAG
14DE DE F8	INCHR3	LDX	XTEMP	RESTORE POINTER
14E0 39		RTS		RETURN
14E1 86 1A	INCHR4	LDA A	#\$1A	SET UP EOF
14E3 20 F1		BRA	INCHR2	

* REWIND FILE (IN RAM)

14E5 DE 97	RWND	LDX	FSTRAM	GET FIRST
14E7 DF 9B		STX	NXTRAM	MAKE NEXT
14E9 39		RTS		RETURN

* STRINGS

14EA 20	CPRSTR	FCC		TSC TEXT PROCESSOR
14EB 20 20				
14ED 54 53				
14EF 43 20				

14F1 54 45
14F3 58 54
14F5 20 50
14F7 52 4F
14F9 43 45
14FB 53 53
14FD 4F 52
14FF 04 FCB 4
1500 43 FCC 'COPYRIGHT (C) 1977 BY TSC'
1501 4F 50
1503 59 52
1505 49 47
1507 48 54
1509 20 28
150B 43 29
150D 20 31
150F 39 37
1511 37 20
1513 42 59
1515 20 54
1517 53 43
1519 04 FCB 4
151A 44 DATSTR FCC 'DATE (MM:DD:YY)? '
151B 41 54
151D 45 20
151F 28 4D
1521 4D 3A
1523 44 44
1525 3A 59
1527 59 29
1529 3F 20
152B 04 FCB 4
152C 54 PRQU FCC 'TYPE P FOR PRINTER... '
152D 59 50
152F 45 20
1531 50 20
1533 46 4F
1535 52 20
1537 50 52
1539 49 4E
153B 54 45
153D 52 2E
153F 2E 2E
1541 20
1542 04 FCB 4
1543 0D CRLFST FCC '\$D,\$R,0,0,0,0,4'
1544 0A 00
1546 00 00
1548 00 04
154A 50 PGSTR FCC 'PAGE LIMITS? '
154B 41 47
154D 43 20
154F 4C 49
1551 4D 49

1553	54	53		
1555	3F	20		
1557	04		FCB	4
1558	07		STPSTR	FCB 7
1559	53			'STOP...'
155A	54	4F		
155C	50	2E		
155E	2E	2E		
1560	07			FCB 7, 4
1561	04			
1562	3F		QUSTR	FCC '?
1563	20			
1564	07			FCB 7, 4
1565	04			
1566	4C		LPPSTR	FCC 'LINES PER SCREEN? '
1567	49	4E		
1569	45	53		
156B	20	50		
156D	45	52		
156F	20	53		
1571	43	52		
1573	45	45		
1575	4E	3F		
1577	20			
1578	04			FCB 4
1579	2A		DVFSTR	FCC '***** MACRO OVERFLOW *****'
157A	2A	2A		
157C	2A	20		
157E	4D	41		
1580	43	52		
1582	4F	20		
1584	4F	56		
1586	45	52		
1588	46	4C		
158A	4F	57		
158C	20	2A		
158E	2A	2A		
1590	2A			
1591	04			FCB 4
1592	2A		BRKSTR	FCC '* PROGRAM BREAK *'
1593	20	50		
1595	52	4F		
1597	47	52		
1599	41	4D		
159B	20	42		
159D	52	45		
159F	41	4B		
15A1	20	2A		
15A3	04			FCB 4

*PRINTER ROUTINES

15A4	39	PRNIT	RTS	*** REPLACE WITH OWN
15A5			RMB	31

15C4 39	PROUCH	RTS		** REPLACE WITH OWN
15C5		RMB	31	

* BUFFER STORAGE AREA

15E4	LINBUF	RMB	155
167F	EXTBUF	RMB	45
16AC	LINBU2	RMB	200
1774	SBUF	RMB	100
17D8	TRAPS	RMB	48
1808	TRPEND	RMB	2
180A	CMNDBF	RMB	180
18BE	TTLBUF	RMB	180
1972	MACTBL	RMB	128
19F2	MTEND	RMB	2
19F4	MACROS	RMB	1534 -
1FF2	LMACRO	RMB	2
1FF4	LAST	RMB	1
1FF5	USER	EQU	*

END

~~TEXT C~~ 7

NO ERROR(S) DETECTED

SYMBOL TABLE:

ADD 005B	ADJS35	0447	ADJS55	0466	ADJSP2	0438	ADJSP3	0444
ADJSP4 0451	ADJSP5	045C	ADJSP6	046A	ADJSP7	0471	ADJSP8	047C
ADJSP9 0483	ADJSPC	0432	APMAC	0DDC	ARB	1184	ATFLG	006C
ATL 0D52	ATL1	0D67	ATL2	0D6A	ATL3	0D77	RTL35	0D7F
ATL4 0D80	ATL45	0D88	ATL5	0D8F	AUTO	00C2	BMPCP	109F
BMPCP2 10A1	BNUM	005D	BRAK	0924	BRKSTR	1592	BTOD	1328
BTOD1 1329	BTOD2	1332	BTOD3	133B	BTOD4	1344	BTOD45	134C
BTOD5 1351	BTODON	13EA	BTOR65	13B0	BTOR92	13DF	BTOR01	1361
BTOR02 136C	BTOR03	137A	BTOR04	1385	BTOR05	1395	BTOR06	13A2
BTOR07 13BB	BTOR08	13CB	BTOR09	13D6	BTOROM	135F	BUFEND	00DE
BUFPNT 00DA	CALMA2	0876	CALMAC	0871	CAP	00B7	CAPIT	07C5
CAPIT2 07CB	CENTE2	0C37	CENTE4	0C3B	CENTER	0C24	CENTJ	05B2
CHKLS2 0F70	CHKLST	0F64	CHKNU2	1232	CHKNU4	123B	CHKNU5	1242
CHKNU6 1246	CHKNUM	121A	CHNG	1112	CHNG2	1126	CHNG25	112F
CHNG3 1130	CHNG4	113C	CHNG45	1144	CHNG5	114D	CHNG6	1160
CLRGET 0689	CLRNUM	119E	CLRSP2	02AC	CLRSP4	02B7	CLRSPC	02A8
CLRTHM 124F	CLSFY	11E6	CLSFY1	11F4	CLSFY2	11FE	CLSFY4	1208
CLSMA2 0F06	CLSMA3	0F08	CLSMA4	0F20	CLSMA5	0F27	CLSMA6	0F2E
CLSMAC 0EF8	CMFLG	0065	CMNDBF	180A	CMNDT	095C	CMMNPNT	00E6
CNJ 00CE	CNTFLG	00D4	CNTRI4	062D	CNTRI5	0633	CNTRI6	0643
CNTRIT 0619	CNTSP2	0610	CNTSP3	0618	CNTSPC	060B	CNTTT2	1075
CNTTT3 1083	CNTTTL	1072	COLCN2	0083	COLCNT	0042	COMAN2	07E5
COMAN3 0805	COMAN4	0815	COMAN5	0818	COMAN6	0824	COMAN7	0828
COMAN8 083C	COMAN9	0848	COMMAND	07DF	CPRSTR	14EA	CRF	00FC
CRLF 1485	CRLFS5	148F	CRLFST	1543	CROM	118D	DATSTR	151A
DAY 0043	DEFM35	0DC2	DEFM45	0DD1	DEFMA2	0DA5	DEFMA3	0DAD
DEFMA4 0DCE	DEFMAS5	0DDB	DEFMAC	0D9E	DELC35	05D3	DELCH3	05CF

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 60

DELCH4 05DC	DELCHR 05B7	DELIM 00BA	DFMFLG 006F	DIVAPP 0E44
DIVER0 0E29	DIVER1 0E2F	DIVER2 0E36	DIVER4 0E3B	DIVERT 0E25
DIVFL2 0092	DIVFLG 0091	DOCAP 0077	DOCM 0078	DONE 006A
DOTAB 0810	DOTAB2 0B1D	DUBB 0C18	DUBH 0C04	DUBH1 0C07
DUBH2 0C0C	DUBW 0C0F	DWFLG 00SE	EBFEND 00E2	ENDLIN 00BF
EV 007E	EXCHR 0062	EXTBUF 167F	FETC22 06F3	FETC25 0706
FETC35 071A	FETC36 071F	FETC37 0722	FETC45 072A	FETC47 0738
FETC48 073B	FETC49 0744	FETC55 0753	FETC57 0767	FETC58 0772
FETC59 0778	FETC65 0787	FETC75 07A5	FETCH2 06DD	FETCH3 070A
FETCH4 0723	FETCH5 0748	FETCH6 077B	FETCH7 079C	FETCH8 07C3
FETCHR 06D4	FILFLG 00C8	FILL 0C82	FINCM 0857	FINCM1 0865
FINCM2 0868	FINCM4 086B	FINIS4 0953	FINISH 0951	FINMAC 0088
FIXBF4 0655	FIXBFE 0646	FIXVA3 1216	FIXVA4 1218	FIXVAL 1209
FIXWD 1414	FIXWD1 1437	FIXWD2 1439	FIXWD3 143F	FLBF 006B
FLUSH 092E	FLUSH2 0939	FLUSH3 0942	FLUSH5 094D	FLUSHB 092A
FNDLS2 0F5A	FNDLST 0F53	FNDMA1 0F3E	FNDMA2 0F4A	FNDMA4 0F58
FNDMA6 0F52	FNDMAC 0F35	FNDNUM 11A8	FNTR 0CF5	FNTR2 0CF0
FNTR4 . 0D11	FNTR5 0D19	FNTR6 0D26	FSTAVL 00A6	FSTRAM 0097
FTCHN1 11CF	FTCHN2 11D2	FTCHNM 11B6	GCNT 0046	GDNUM 005A
GETC22 069E	GETC25 06A7	GETCH1 0693	GETCH2 069A	GETCH3 06AB
GETCH4 06BB	GETCH5 06C4	GETCH6 06D1	GETCHR 068C	GETIN 0B75
GIBUF 1444	GIBUF2 1448	GIBUF4 145F	GIBUF6 1469	GTNA4 0FAB
GTNA6 0FB0	GTNAM 0F8A	HIPG 0096	IF 10A5	IF1 10A8
IF3 10B6	IF35 10BC	IF4 10C7	IF5 10D4	IF55 10E6
IF6 10F2	IF8 1105	IFFLG 008A	IFN 10ED	IFN2 10F1
IFY 10D0	INC 005F	INCH 0206	INCH15 14CB	INCHR 14BA
INCHR0 14C3	INCHR1 14C8	INCHR2 14D6	INCHR3 14DE	INCHR4 14E1
IND 0048	IND2 0084	INDEX 00F6	INDNT 0ADB	INDNT2 0AEB
INIT 02C0	INIT25 02C4	INIT3 02FB	INIT4 031A	INMAC 0EE3
INMAC2 0EE5	INMAC4 0EF3	INMAC5 0EF7	INNUM 0070	INSS44 0604
INSSP2 05E7	INSSP3 05ED	INSSP4 0601	INSSP5 060A	INSSPC 05DD
INTRO 0212	INTR00 0219	INTR03 0259	INTR04 026A	INTRO5 027F
INTR06 029D	INTRP 0D97	JNKCNT 009D	JST 0AA1	JST1 0AB1
JST15 0AB6	JST2 0AB7	JST3 0AC1	JST4 0ACA	JSTF25 03AF
JSTF55 03D3	JSTF63 03E0	JSTF65 03EB	JSTF95 041D	JSTFY 0397
JSTFY1 03A8	JSTFY2 03AD	JSTFY3 03B8	JSTFY4 03C8	JSTFY5 03D1
JSTFY6 03D6	JSTFY7 03F0	JSTFY8 03FA	JSTFY9 0412	JUST 00D6
LAST 1FF4	LDIV 0055	LDNSK2 11DA	LDNSK4 11E3	LDNSKP 11D8
LEFT 006D	LEFTM 0ACB	LEFTM1 0AD8	LEFTM2 0ADA	LENT25 0B05
LENTH 0AF1	LENTH2 0B01	LENTH5 0B0F	LFM 004E	LINBU2 16AC
LINBUF 15E4	LINCNT 004D	LINS 048F	LINS2 0494	LINS3 049F
LINS4 04AC	LINS5 04B6	LINS6 04B8	LINS7 04C5	LLN 004B
LLN2 009F	LMACRO 1FF2	LOWPG 0095	LPPSTR 1566	LSTAVL 00A4
LSTNUM 0063	LSTRAM 0099	MACCNT 008B	MACEND 00FA	MACNAM 00B0
MACOVF 084E	MACROS 19F4	MACTBL 1972	MACTMP 00A2	MBFLG 0066
MBFPNT 0067	MCEND 0F7D	MCEND2 0F87	MCNT 008C	MINDIS 007D
MNTH 004C	MON 0209	MSC 00D2	MTEND 19F2	MULTS 0A8B
MULT32 0A94	MULT33 0A96	NEDL 0CE6	NEDL1 0CEE	NEDL2 0CF0
NEDL4 0CF4	NEG 0071	NEGT 0089	NMREGS 0040	NOCAP 0C72
NOCR 0069	NOEXP 007F	NOFILL 0CAB	NOFL 006F	NOJST 0A9D
NONUMS 008D	NOOUT 0079	NOSPC 0B23	NPGN 005E	NREG 1165
NREG4 1183	NSP 0073	NUM 0125	NUMPNT 0060	NXTMAC 00AC
NXTOUT 00AE	NXTRAM 009B	NXTTAB 0080	NXTTRP 0085	OPAPP 0E82
OPAPP2 0E8A	OPAPP4 0E8F	OPMAC 0E4F	OPMAC2 0E57	OPMAC4 0E60

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 61

DPMAC5 0E6C	OUTCH 0203	OUTCH2 14A9	OUTCH3 14AE	OUTCH4 14B7
OUTCHR 149F	OUTL55 0546	OUTL75 056E	OUTL82 0595	OUTL85 059F
OUTLI1 0517	OUTLI2 0520	OUTLI3 052A	OUTLI4 0532	OUTLI5 053F
OUTLI6 0550	OUTLI7 056B	OUTLI8 0589	OUTLI9 05A4	OUTLIN 05B8
OUTM18 0ECF	OUTMA0 0EB9	OUTMA1 0EBB	OUTMA2 0ED1	OUTMA3 0ED2
OUTMA4 0ED7	OUTMAC 0ERE	OUTSV 0D49	OVFSTR 1579	PAGE 0A67
PAGE2 0A73	PAGE4 0A7E	PAGES 0A83	PAGE6 0A8A	PAGEL 0C44
PAGEL1 0C4D	PRGEL2 0C56	PAGEL4 0C58	PASCHR 0075	PASFLG 008C
PASS 0A3D	PCHAR 00CC	PCRLF 0899	PCRLF2 08A8	PCRLF4 08AD
PDATA 1492	PDATA2 149E	PFLG 00CA	PGL 004F	PGN 0074
PGNUM 0BF7	PGNUM4 0C03	PGSTR 154A	PINIT 020C	POUCH 020F
PRNIT 15A4	PRNTR 0093	PRNU27 125E	PRNU28 1272	PRNU31 127E
PRNU32 1288	PRNU35 1297	PRNU37 129F	PRNU45 12C4	PRNU65 12EA
PRNU72 1303	PRNU73 1307	PRNU82 1319	PRNUM 1256	PRNUM2 1258
PRNUM3 1277	PRNUM4 12A7	PRNUM5 12CC	PRNUM6 12E0	PRNUM7 12F0
PRNUM8 130F	PROC 033D	PROC2 0343	PROC3 0349	PROC4 0350
PROUCH 15C4	PRQU 152C	PSTRNG 1490	PTFL 0078	PTIND 0C76
PTIND2 0C83	PTIND3 0C98	PTIND4 0CA4	PTIND5 0CRA	PULLX 13FF
PUNT35 036A	PUNTS2 035B	PUNTS3 0360	PUNTS4 0382	PUNTS5 0385
PUNTS6 0387	PUNTS7 038D	PUNTST 0353	PUSH4 13FB	PUSHX 13ED
QUSTR 1562	REMMA4 0DEE	REMMA6 0E0B	REMMAC 0DE5	REMNAME 0E15
RESPC 0B26	RETMA2 0678	RETMAC 066D	RETREG 00F4	RIGHT2 05AA
RIGHTJ 05A7	RINS 04C9	RINS2 04CD	RINS3 04D4	RINS4 04DF
RINS5 04ED	RINS6 04F7	RINS7 04F9	RINS8 0507	ROM 00C4
ROM2 118A	RPT 0CB9	RTJ 00D0	RWND 14E5	SAUTO 1191
SAUT04 119D	SAVS 0D2D	SAVS1 0D38	SAVS2 0D3A	SAVS25 0D3E
SAVS4 0D44	SAVS5 0D48	SAVSX 0EA0	SAVSX2 0EAB	SBFLG 009E
SBUF 1774	SCAP 00B8	SCRL55 08E8	SCRL75 0906	SCRL85 091A
SCRLF 08B1	SCRLF1 08B4	SCRLF3 08C0	SCRLF4 08DC	SCRLF5 08E3
SCRLF6 08F4	SCRLF7 0903	SCRLF8 0910	SCRLF9 0923	SENV 0B33
SENV1 0B90	SENV2 0B91	SENV3 0B96	SENV4 0BB3	SENV6 0BC2
SENV8 0BDE	SENV9 0BED	SENV95 0BF0	SETCA2 07DB	SETCAP 07CE
SIGN 0072	SIN 007C	SIND 0C59	SNGLS 0A99	SPACE 0A44
SPACE2 0A55	SPACE4 0A5E	SPACE6 0A66	SPCPT1 00E8	SPCPT2 00EA
SPIFLG 0090	SPSPF 0076	SROM 1188	STAB 0B41	STAB2 0B44
STAB4 0B5A	STACK 01FF	START 0200	STCRP 0C6D	STOP 0CD2
STOP1 0CD8	STOP2 0CE5	STPOUT 00A8	STPSTR 1558	SUB 005C
SUPL 00B5	SVDSPC 0087	SWRDF 00B6	SYERR 0EE0	TAB 00C0
TABCH 0B2A	TABCH2 0B32	TABEND 0124	TABFI2 0B3E	TABFIL 0B35
TABFLG 0082	TABS 0110	TBLEND 0A3C	TCNT 00BB	TCPNT 00AA
TEMP 00EC	TEMP2 00EE	TEMPS 00F0	TEMP6 00F2	TERM 0B60
TERM2 0B63	TERM4 0B6C	TFILF 006E	TFILL 00C1	TIND 00B3
TITL12 0FE1	TITL15 0FE7	TITL65 1041	TITL75 1056	TITL76 1060
TITLE 0FC1	TITLE1 0FD1	TITLE2 1001	TITLE4 1024	TITLE5 102F
TITLE6 1032	TITLE7 1049	TITLE8 1063	TITLE9 106E	TLEN 0FB4
TLEN2 0FC0	TLLN 00B4	TLN 00D8	TLPP 0094	TOUTL 007A
TPOS 00B9	TRAPS 17D8	TRPEND 1808	TSIN 00B2	TSTBR2 1477
TSTBRK 1471	TSTNE2 1111	TSTNEG 1106	TTLBUF 18BE	TTLPTN 00BD
USER 1FF5	WIDTH 00C6	XFRRT2 109D	XFRRTL 1086	XMAC 00B0
XTEMP 00F8	YEAR 0058			