

RIPOFF

Another Text Formatting Program

by

Donald P McKay and Gerard Donlon

contributions by

Hal Shubin and Joao Martins

SNeRG
Technical Note No. 5
Department of Computer Science
SUNY/Buffalo
4226 Ridge Lea, Amherst, NY, 14226
July, 1980
Revised July, 1982

1. Introduction
 2. Formatting with RIPOFF
 - 2.1 The basics
 - 2.2 Page features
 - 2.3 Paragraph features
 - 2.4 Line features
 - 2.5 Character features
 - 2.6 Miscellaneous features
 3. Ripoff and the Malibu Printer
 - 3.1 CNTL F Font Changes
 - 3.2 CNTL C Special Characters on the MALIBU
 - 3.3 The MALIBU Command
 - 3.4 MALIBU as the Output File
 - 3.5 The MALIBU's Printing speeds
 4. Extending RIPOFF
 5. Mea culpa, mea culpa
- Appendices
- I. RIPOFF Command Summary

1. Introduction

RIPOFF is a text formatting system written in ALISP which runs on the CYBER 174 at SUNY/Buffalo. RIPOFF takes a source file of mixed text and commands and produces an output file which is a formatted version of the source file. In general, the source file consists of an initial set of RIPOFF commands which declare properties of the formatted document which are different from the defaults provided by RIPOFF followed by the text to be formatted. Commands generally appear on separate lines which have '\$' (a single dollar sign) as the first character of the line. The commands must be in upper case (this feature may be eliminated in the future). The text to be formatted appears in mixed upper/lower case. Any editor which can edit ASCII (upper/lower case) files may be used to edit RIPOFF source files.

Three types of output files are available. The standard disk output file assumes an ASCII printer, one that accepts FORM FEED, CARRIAGE RETURN, LINE FEED, etc. and does the appropriate thing with them. The second type of output file is TTY format. This is the standard format when the output file TTY is selected. TTY format utilizes features of the NEC terminals available for departmental use in the Department of Computer Science. The third type of output file is MALIBU format. This is the standard format when the output file MALIBU is selected. MALIBU format utilizes features of the MALIBU printer and the ADM-31 terminal

The basic formatting philosophy is to provide a rich set of defaults which correspond to the standard formatted document, e.g. double spacing, left margin at 10 characters (1 inch), page size of 85 characters by 66 lines (8 1/2 X 11 inch paper), etc. Thus, RIPOFF will be quite painless to use for research papers or technical notes. As the number of commands required by a document increases RIPOFF becomes a little clumsy to use -- more commands and their interactions have to be remembered.

This description of RIPOFF discusses basic formatting issues, some advanced features, how to extend the system and some weaknesses of the current implementation. It assumes some knowledge of ALISP.

2. Formatting with RIPOFF

2.1. The basics

To run RIPOFF, use the following Cyber control statements:

```
ATTACH,ALISP/UN=CSDLIB.
ALISP,FL.
(Load (RIPCOM CSDLIB))
```

The top level of RIPOFF interacts with the user to determine if she wishes to create, edit, print, or extend a file. It then asks for the name of the source file. Type a file name, i.e. a legal argument to the ALISP function OPEN. It next asks for an output file. If the output file is TTY or MALIBU then RIPOFF will use the terminal. If the output file is not TTY or MALIBU then it is an ALISP file name; a legal file name for CLOSE. When RIPOFF prints a new page to any file but TTY or MALIBU, it prints a "." to the terminal to let you know its actually doing some work (although this feature is essentially nullified by the buffering in the communications multiplexor). Here are several sample interactions with RIPOFF.

```
TYPE C[CREATE],A[ADD],E[EDIT],P[PRINT],S[END],Q[UIT],L[ISP] OR H[HELP] ?P
```

```
AUGUST 20, 1981      3:24 PM
SOURCE FILE
?RIPTTEST
OUTPUT FILE
?RIPOUT
```

```
BEGINNING RIPOFF RUN ...
```

```
... RIPOFF RUN ENDS -- 1 pages in RIPOUT
TYPE C[CREATE],A[ADD],E[EDIT],P[PRINT],S[END],Q[UIT],L[ISP] OR H[HELP] ?E
FILE NAME
?RIPTTEST
WHICH EDITOR DO YOU WANT TO USE (BED VW OR UCEDIT)?U
```

```
<some interactions with UCEDIT>
```

```
TYPE C[CREATE],A[ADD],E[EDIT],P[PRINT],S[END],Q[UIT],L[ISP] OR H[HELP] ?Q
<RIPOFF exits ALISP>
```

Note that create and add commands use the TELEX TEXT command, edit will ask you which editor you want to use the first time you edit in a ripoff session, after that it remembers which editor to use. Help has not yet been implemented, lisp will exit ripoff and leave you in alisp. and send copies a file to your terminal at the requested speed.

Before introducing any RIPOFF commands, the format for commands is described. A command line is distinguished by a single dollar sign (\$) appearing in the first character position of a line. Most of the commands discussed here are called in this way. After the dollar sign, a single command expression indicates which

command(s) is(are) to be invoked. Note, since the commands are actually LISP functions, the command expression indicates a function to be called. The command expression may be an atom in which case the function of no arguments by that name will be called. If the command expression is a list of atoms the expression is treated as a single function call. Otherwise, the command expression is a list of function calls (note a list of atoms is not considered a list of function calls). For example, the command expression "\$PAGE" invokes the PAGE command, a function of no arguments. The command expression "\$(LEFTMARGIN 15)" sets the left margin to the 15th character position. The command expression "\$((PAGE)(LEFTMARGIN 15))" first does a PAGE and then sets the left margin. The remainder of this section discusses features of pages, paragraphs, lines and characters. The descriptions will be rather brief, further documentation of the commands appears in Appendix I.

2.2. Page Features

A formatted page is printed on a space which is defined by the number of lines per page and the number of character positions available on a line. The command which declares this is PAGESIZE. The default is 66 lines and 85 characters per line (8 1/2 X 11 inch sheets). This is sufficient for most documents, you need to change the PAGESIZE for different page sizes and different typing element sizes (e.g. 12 characters per inch or 88 lines per page). On the declared page, formatted text will appear only between the left and right margins. The LEFTMARGIN and RIGHTMARGIN commands reset these. The defaults are a 1 inch margin on both sides, i.e. a LEFTMARGIN of 10 and a RIGHTMARGIN of 75. There is also a margin for the bottom of the page. BOTTOMMARGIN declares the number of blank lines to be left at the bottom of each page; defaults to 5.

The top of each page can contain a title and/or page numbering. The TITLE command takes a string which will be printed at the top of each succeeding page. If the title is NIL then no title appears; the default is NIL. If page numbering is desired the PAGENUM switch must be T; default is T. Every page except the first is numbered. (For every command which assigns a value to a switch, either ON or T can be used to select a feature and either OFF or NIL deselects it.) The command PAGE-NUMBER with a number argument sets the page number to that number. The command PAGE-LABEL allows more complicated page numbering, for example, it allows pages to be numbered I.1 I.2 ... and then you can change to II.n for the second section, and so on. The PAUSE command with argument ON declares that if the output is to the file TTY or if the TTY format has been selected then RIPOFF will pause between pages. It prints a bell to indicate that it is done with the current page and that a new page should be inserted. When a new piece of paper is

inserted type a carriage return to continue processing. The output will start on the current line. The default is PAUSE OFF. The default should be used for continuous forms.

The PAGE command forces the current page to be terminated and a new page to be started.

The LINES-PER-INCH command declares the number of lines there are in an inch of output. This is required for superscripting and subscripting discussed below. It defaults to 6. If you use 8 lines per inch and want to do superscripting or subscripting you have to indicate to the system that the default does not hold.

Two commands are predicates about where processing is on an output page. The predicate TP returns T if there are at least as many lines remaining on the current page as its argument. The predicate TOP-OF-PAGE returns true if processing is at the beginning of a fresh output page. These are useful for extending RIPOFF by user defined functions (see below).

There is a model page capability for printing papers for camera ready copy of conference proceedings, etc. See Section 2.6.

2.3. Paragraph features

The PARAGRAPH command terminates the current paragraph, skips the number of lines specified by the PARAGRAPH-SKIP and indents the number a spaces declared in the PARAGRAPH-INDENT command. The PARAGRAPH-SKIP defaults to 1 extra line between the end of one paragraph and the start of another. The default for PARAGRAPH-INDENT is 5 spaces. Note that this is in addition to any INDENT or LEFTMARGIN that may be in effect.

To make paragraphs trivial to start and end, the AUTOPARAGRAPH command sets a switch to control automatic paragraphing. If its value is ON then any line which starts with at least one blank signals the end of the current paragraph and the start of the next one. If AUTOPARAGRAPH is OFF then there will be no such processing. The default is AUTOPARAGRAPH ON. AUTOPARAGRAPH must usually be set to OFF for tables or other similar types of unformatted text (see the TRANSPARENT command below).

Numbered paragraphs or numbered lists can be generated using the ITERATE and END-ITERATE command. The ITERATE command takes three arguments, an indentation a paragraph skip count and the character or number to start iterating with. The paragraphs following the ITERATE command and before the END-ITERATE command will be labeled sequentially. If the iteration character is not a number, then the

next character used will be the next character in the alisp character set. If the iteration character is a number then the next character used will be the next integer. The default extra indentation is 0, the default paragraph skip is 1, and the default iteration character is 1, i.e. numbered iteration starting at 1. This command is useful for generating numbered reference lists.

The REFERENCE command is just like the ITERATE command but no numbering is done. The command END-REFERENCE terminates the REFERENCE command. REFERENCE temporarily resets PUNCTUATION to none until the END-REFERENCE command. Also, the word "Reference" is centered on the output page before the first paragraph is output. Most useful in generating reference lists indexed by some other way than numbers. If the atom NOPAGE is the third argument then the reference list does not start on a new page. If NOPAGE is not present then the reference list starts at the top of a new page.

2.4. Line features

Each output line starts in character position LEFTMARGIN + INDENT. The INDENT command specifies how many extra spaces are to appear before the first non-blank character. It defaults to 0. INDENT sets indentation to its argument, i.e. it's an absolute indentation. When an output line is terminated (other than by BR or SKIP) the current SPACING number of lines are skipped. So, an argument of 2 for SPACING yields double spacing, 1 single spacing, etc. The default SPACING is 2.

The BR command of no arguments terminates the current output line without regard to how many characters appear on the line or how the output items are positioned on a line (see FILL and JUSTIFY commands below). The SKIP command skips the number of lines specified by its argument. It does a BR first.

The commands FILL, JUSTIFY and TRANSPARENT select how many items will appear on an output line and whether there is a ragged right margin. The FILL command with argument ON selects filling output lines to capacity, i.e. words will be put onto an output line until the addition of another word would exceed the right margin. The default is FILL ON. If FILL is OFF then the end of the input line signals the end of an output line. If the JUSTIFY switch is ON (and FILL is ON) then after the output line has been formed it is manipulated to ensure that the last character of the last word is placed on the right margin. Blanks are interspersed between words as necessary. The default is JUSTIFY OFF -- ragged right margin. The TRANSPARENT command allows easy selection of standard and tabular paragraph and line features. If its argument is ON it turns off FILL,

JUSTIFY and AUTOPARAGRAPH, i.e. the output line is exactly the same as the input line, no formatting is performed. If its argument is OFF it restores the previous values of FILL, JUSTIFY and AUTOPARAGRAPH.

The CENTER command takes a line of text as an argument. It centers the line of text following the CENTER command on the next output line after terminating the current output line with BR. For example,

```
$CENTER
```

```
A line of text
```

results in the output

```
A line of text.
```

2.5. Character features

Every output item which ends in one of ".", "!", ":", ";", or "?" has at least two blanks appear after it regardless of the FILL and JUSTIFY switches. The PUNCTUATION command takes a list of characters for which this is to be done. One can turn off all special punctuation by calling the PUNCTUATION command with no argument. This is useful for reference lists which contain extraneous punctuation.

The UNDERLINE command selects the character used for underlining. It defaults to underscore, _. If the UNDERLINE character is not underscore then underlining is done on a separate line. This is required on line printers which do not have underscore, for example the Department's Centronix printer. Otherwise, if the UNDERLINE character is underscore, underlining is done by overprinting on the line. This command interacts with the @U macro character. To underline some item(s), enclose it in @U's. Note that all characters appearing between the @U's are underlined and that FILL and JUSTIFY can effect how many blanks might be underlined.

When output is to the NEC terminal or MALIBU printer, a limited superscript-subscript feature is available. Note, this means the TTY or MALIBU format is selected either by the TTY or MALIBU command or the TTY or MALIBU output file. The commands @A shift to Above the current line. The commands @B shift to Below the current line. To write a subscript, type @B<stuff to appear as subscript>@A. The @B moves down a half line and whatever text appears between it and the @A is placed a half line below the normal line. The final @A resets the printer to the normal spacing line. Superscripts are just the reverse. Type @A to move up a half line, the superscript text and finish with a @B. Subscripts and

superscripts cannot be extended over more than one line.

Several commands manipulate the position of the typing element on the NEC and other terminals. The @R command moves in the Reverse direction on the line one space, i.e. its a backspace. The @S command results in a forced Space. In other words, every @S in the input is treated as an output item which prints as a blank.

To have an @ appear in the output file use @@ in the source file.

2.6. Miscellaneous features

The FIGURE command reserves space for figures, supplies a title, optionally boxes the figure to set it off from the surrounding text and optionally specifies a file which is to be output in TRANSPARENT mode (see above). FIGURE reads its arguments from the remainder of the line on which it appears and also the next line. The second line is the caption for the figure and will be centered under the figure identifier. For example,

```
$FIGURE 10
```

```
A one line title
```

results in ten blank lines being reserved on the next page which has at least 10 lines remaining. The command

```
$FIGURE 14 FIG1
```

```
Another one line title
```

will read from the file FIG1 when the next page has at least 14 lines on it. The FIGURE command reports an error if it is impossible to place a figure on a page because the declared length is greater than the page length between the first printable line and the bottom margin (see BOTTOMMARGIN above). RIPOFF command characters can be used in figures.

An option for a "box" around a figure is available. It will draw solid lines above and below a figure. To use it, call BOXES with ON at the beginning of the source file. The default is BOXES OFF. BOXES should be set for an entire document and not reset in the middle of a document.

When a figure is printed, the body of the figure is output followed by a figure designation and the figure title. The figure designator is by default just the word "Figure" followed by a number, e.g. "Figure 1". If FIG-LABEL is called with any atom then the atom becomes prefixed to the number, e.g. if the figure label has been set to A, then the first figure is designated "Figure A.1".

The command FIGURENUM sets the current figure number to its argument. The first figure defaults to 1.

The NEWSECTION command leaves space for a section title. It leaves a space before the title and makes sure that the section title will not be the last thing on a page. You can either print your own section title or use the SECTION command after the call to NEWSECTION.

The SECTION command allows numbering of sections to three levels. The command call is \$(SECTION n title) or \$(SECTION n title NOPAGE) where n is the level of section requested, i.e. 1 for a section, 2 for a subsection and 3 for a sub-subsection and title is a string enclosed in double quotes which will appear after the section number and if n is 1 will also appear as the title on pages of the section. For every level 1 section, the ENDSECTION command is called automatically (see below) without the NOPAGE option unless NOPAGE appears in the command, i.e. a new level 1 section starts at the top of a page unless you specify NOPAGE. For the other levels, ENDSECTION is called with the NOPAGE option. There are, however, some extra blank lines used on either side of the section heading in the text. Figures within a section are labelled with the section number as in FIG-LABEL.

The ENDSECTION command flushes all delayed figures from the system. It is useful when it is desired to have all figures associated with a particular section of a paper appear before the start of the next section. ENDSECTION also does a PAGE after all delayed figures have been output if NOPAGE is not an argument to ENDSECTION.

A model page mode is available for printing a paper submitted to a conference proceedings or journal article. The MODEL-PAGE command takes several arguments which define the characteristics of the model page format. They are described in more detail in Appendix I. Essentially, the model page mode prints narrow columns of text which can be pasted onto the model paper. Figures can be declared to be reduced in size and columns can be restricted so that figures only take up a certain percentage of the space in a column. Figures are not in the output file but in a separate file named FIGFILE which is overwritten for each RIPOFF run using MODEL-PAGE. The command also turns JUSTIFY ON.

3. Ripoff and the Malibu Printer

3.1. CNTL F Font Changes

@Fx with x=[1,2,3,4,5,6,8,D,L,Q,N] will change the font as specified by x. If x is a number, change to that font number. If x is D go to draft quality. If x is L go to letter quality. If x is Q change back to the letter quality active before the last @FL or @FD. If x is N change back to the font number active before the last @Fn for n a number.

Font 1 (@F1) is the standard font and the default quality is Draft. Ripoff does not enforce these defaults at the start of printing. Ripoff will use whatever the printer is set at as the starting font and quality. If the first font change commands are @FQ @FN then the printer will use font 1 in draft quality. To print the final version of a paper, include the letter L in the call to malibu, or put @FL immediately before the first printable character. Then, as long as there are not more @FQ's than @FL's and @FD's, the paper will be printed in letter quality.

Font 4 has the same letters as font 1 but font 4 is smaller. By switching fonts you can sub-super script with small letters. Font 2 in letter mode has even smaller numbers giving X^2 and X_0 printed using X@F2@FL2@FN@FQ and X@F2@FL@BO@A@FQ@FN. Only the numbers in font 2 are small. The rest of font 2 is Greek.

You can switch Letter Quality anywhere you want. You can also switch fonts anywhere you want to. You can even print in Αθεεκ or α*□ (Greek and APL).

NOTE that if you print a fixed number of characters per line, as RIPOFF does, then changing to a smaller print sizes will move the right most character over to the left. Changing to a larger print size will move the right most character farther to the right.

NOTE ALSO: there is no character set numbered 7 and character set number eight is for down-loaded characters only.

3.2. Cntl C Special Characters on the Malibu

Several special characters have been written for the Malibu. When output is in MALIBU format, RIPOFF will print the special characters. When output is not in MALIBU format, RIPOFF will print the characters name.

The characters currently available are ANDOR, THRESH, NONDERIVABLE, DERIVABLE, EXISTS, NEXISTS FORALL(or ALL), IMPLIES(or =>), ORIMPLIES(or V=>), ANDIMPLIES(or &=>), EQUIV, ENTAILS, and DELTA.

To print a universal quantifier, existential quantifier, delta, entails, derivable, nonderivable, equiv, or any of the implies, include @C<symbol name> in the text, as in @C FORALLx which prints as $\forall x$. note that any number of blanks can separate the @C from the symbol name but there can be no blanks in the symbol name. Normal printing starts immediately after the symbol name.

The other symbols require an additional argument, a list of subscripts and superscripts to accompany the symbol. For instance, @CANDOR (0 n-1 n) will print $n \times \theta^{-1}$. The arguments are, in order, the minimum, maximum, and total count. Any or all of the arguments may be omitted by replacing it with NIL or OFF. () can be used when there are no arguments. Any number of blanks can separate the symbol name from the @C and the argument list. Normal printing resumes immediately after the argument list So, to print an andor alone, use @CANDOR () which prints a \times .

3.3. The MALIBU Command

The MALIBU command causes ripoff output to be formatted for printing on a Malibu dual-mode 200 printer.

The format of the call is

\$(MALIBU sw)

OR

\$(MALIBU ON f q)

If sw is T or ON then output is formatted for the Malibu. If f or q are a font number or letter quality (i.e. 1-6,8,L or D) then ripoff sets the MALIBU to that font number and letter quality. The defaults are 1 and D. If the fonts are not specified, ripoff will not set the font on the printer, that way, a paper that does not use any font changes or special characters (see previous section) can change from draft to letter quality by setting the printer. Other papers must be run through ripoff again.

This command should be used in the first sequence of RIPOFF commands, before any text.

If the output file is specified as MALIBU then the MALIBU command has no effect. Malibu is initially NIL, OFF.

3.4. MALIBU as the Output File

If MALIBU is specified as the output file then RIPOFF will format its output for immediate transmission to a Malibu dual-mode 200 printer through the printer port of an ADM-31 terminal. This is usefull for some draft copies but should not be used for letter quality papers as the printer will not be able to keep up with RIPOFF in letter quality mode.

3.5. The Malibu's Printing Speeds

The Malibu can print draft quality papers at between 165 and 250 characters per second. And letter quality at between 42 and 60 characters per second, depending on which font is being used and as long as it is only printing characters. If the printer has to print subscripts or superscripts, over print, or do form feeds or repeated line feeds, the printer speeds drop to 90 to 120 for draft quality and 30 to 55 cps for letter quality.

To make this problem less bothersome, RIPOFF will print null characters after each subscript, superscript, overprint and form feed. This makes RIPOFF output files larger but, since the printer ignores nulls, it allows you to send draft quality papers at 120 cps and letter quality papers at 42 to 60 cps depending only on the font used. To stop printing nulls, use the command (NULLS OFF). For more control over the number of nulls printed, see the command NULLFACTOR.

The Malibu does have a 512 character buffer.

4. Extending RIPOFF

As with any nontrivial program, making modifications to RIPOFF can be very simple or very difficult. Simple additions, such as adding an indexing facility, which are independent of any part of the current system can be made quickly. Any modification which requires substantial interfacing to existing code could be more of a problem. Here an example would be the addition of graph printing, which depending on how sophisticated the implementation, could require a rewrite of the entire print routine.

There are three ways to make additions to RIPOFF:

1. using LISP expressions,
2. defining LISP functions which call RIPOFF functions,
3. adding new RIPOFF macro characters.

One way to extend RIPOFF is to use LISP expressions containing calls to RIPOFF functions. For example, a command which both tests for a number of lines remaining on a page and if sufficient lines are not available starts a new page does not exist in the current implementation. The command expression "\$ (IF (NOT (TP 10)) (PAGE))" tests for at least 10 lines remaining on a page. Similarly, the command expression "\$ (IF (NOT (TOP-OF-PAGE)) (PAGE))" guarantees that formatting begins at the top of a page.

Another way to extend RIPOFF is to define a new command, i.e. function. By combining existing functions, one can extend RIPOFF in a limited way. For example, a command which both tests for a number of lines remaining on a page and if sufficient lines are not available does a PAGE does not exist in the current implementation. One can define such a function using the existing RIPOFF commands. The function could be defined as follows:

```
(DE NEED (N)
  (IF (NOT (TP N))(PAGE)))
```

As another example, consider a command to guarantee that formatting begins at the top of a page. A new function could be defined in the following way:

```
(DE FORCE-PAGE NIL
  (IF (NOT (TOP-OF-PAGE))(PAGE)))
```

An extension of this type that might be nice is something similar to SCRIBE's document types. For example, a command which declares the appropriate format for memos, letters, tech reports, journal articles, etc.

A final way to extend RIPOFF is thru the use of RIPOFF macro characters. RIPOFF macro characters are characters for which there exists a function of no arguments under the property RIPMAC (see DRM in the Appendix and the source code for RIPOFF, especially @). The command character, @, mentioned above is implemented as a RIPOFF macro character. For example, if you want to use the limited font capability on the NEC or if there is some printing device with multiple fonts then defining a new font change macro character or extending @ would be the easiest approach.

5. Mea culpa, mea culpa

There are obvious features omitted which are absolutely necessary for RIPOFF to become more than a tool for our research group or even to be used extensively by more people than the authors. Also, the implementation has been hacked -- anytime something needed to be fixed the pragmatic way out was chosen. This section mentions some of the missing features.

The top level of RIPOFF does not impose a structure on documents. Thus, the source file is a single file and figure files are separate. What this means is that one large file must be completely processed and several small files must be present for a successful RIPOFF run. Its a burden on the user to remember all the little files. What would be nice is an approach similar to APTEDIT's which stores all text for a document in one source file but can split it up into smaller "chapters" or "sections".

Currently, there is little font processing. Unless you are using a Malibu printer you're stuck with the typing element with which you start the RIPOFF run.

There is no indexing facility.

The help facility hasn't been written.

Commands are upper case only and there are no abbreviations for them.

RIPNEW 11 AUGUST 1981 13.49.17
 CREATED: 15 APRIL 1981 21.52.49
 LAST MODIFIED: 11 AUGUST 1981 13.44.47
 CHANGES MADE TO: LINES-PER-INCH SKIP

/@ NIL NIL

A RIPOFF read macro which performs special character processing depending on the character following the @. All command characters following @ may be either upper or lower case.

- @@ - results in a @ in the output; @ is treated as an escape character, so @@ is the way to get a single @.
- @A - moves the printing element up one-half line; @A is used to start superscripting and terminate subscripting, see @B.
- @B - moves the printing element down one-half line; @B is used to start subscripting and terminate superscripting, see @A.
- @C - print a special character on the malibu or a blank if if output is not in malibu format. Characters currently available are: ANDOR, THRESH, DERIVABLE, NONDERIVABLE, IMPLIES(or =>), ORIMPLIES(or V=>), ANDIMPLIES(or &=>), EXISTS, NEXISTS, FORALL(or ALL), EQUIV, and DELTA.
- @F - change fonts on the malibu.
 Has no effect if output is not in MALIBU format. @Fx changes fonts: if x is a number change to that font if x is L change to letter quality, if x is D change to draft quality. if x is N change back to the font number active before the last @Fi for i a number, if x is Q change back to the letter quality active before the last @FL or @FD.
- @R - moves the printing element back one character space; @R is useful for overprinting.
- @S - forced space character; is treated as any other printing character.
- @U - signals underlining either to begin or end; @U brackets the text to be underlined, eg @Uunderline this@U.

AUTOPARAGRAPH LAMBDA (SW)

This command takes either T or NIL as its single argument. If SW is T, then autoparagraphing will take place, otherwise it will not. Autoparagraphing mode will start a new paragraph whenever a line of text is encountered which has a blank as the first character on the line.

BOTTOMMARGIN FLAMBDA (N)

This command sets the number of lines which will be left blank at the bottom of each page. For example, (BOTTOMMARGIN 5) will reserve 5 lines at the bottom of each page. Initially set to 5.

BOXES FLAMBDA (FLG)

BOXES used to turn boxes drawn around figures on or off.
 ----- The switch %FIGBOX will be set to the argument of this function, which should be T or NIL, unless new uses are created. -- hs
 NOTE: USE THIS FN ONLY AT THE BEGINNING OF A FILE, OTHERWISE YOU MAY HAVE A FIGURE WITH 3 LINES TOO MANY OR TOO FEW.

BR LAMBDA NIL

This command terminates the current line. Note that all it does is end a line, it does take into account the spacing factor that may be in force.

CENTER LAMBDA NIL

This command is a two line command. The next input line after the command is a piece of text which will be centered on the next output line. Almost equivalent to the command list ((BR)(SKIP 1) ... <center the output> ... (SKIP <spacing-factor>)).

ex -
 \$CENTER
 A piece of text.

DRM FLAMBDA DEF

Defines a RIPOFF read macro character.

END-FIGURE LAMBDA NIL

Ends a FIGURE INLINE command. Can be used in OFFLINE figures to end the contents of the figure before the end of the text.

END-ITERATE LAMBDA NIL

Ends an ITERATE command.

END-REFERENCE LAMBDA NIL

Terminates a REFERENCE command.

ENDSECTION FLAMBDA ARGS

Forces all delayed figures to be printed and resumes processing at the top of a new page only if NOPAGE is not in ARGS. Intended to "flush" a section of a paper.

FIG-LABEL FLAMBDA (LABEL)

Labels a figure with LABEL such that LABEL.n is the figure number.

FIGURE LAMBDA NIL

This command has two forms. The first does not process the contents of the figure. The second does.

This form reserves space for figures. In syntax it is similar to the center command. The command has two arguments, the number of lines to reserve and a one line title. The number of lines is in absolute terms, ie. spacing factor is ignored. Equivalent to the command stream ((BR) (SKIP <n>) ... (CENTER <next input line>)(SKIP <spacing factor>)).

ex -
 \$FIGURE 15
 A one line title.

The FIGURE command can also take a file name as argument which specifies a file which will be copied as the body of the figure. In this case the number of lines parameter indicates the size (number of lines) in the specified file.

ex -
 \$FIGURE 10 FIG1
 A one line title.

The second form of the FIGURE command formats and adds the figure. The contents of the figure may be on a separate file or included in the text. Figures included in the text are terminated with a call to the function END-FIGURE.

ex. of an inline figure
 \$FIGURE INLINE
 A one line title.
 the figure ...
 \$END-FIGURE

ex. of an offline figure
 \$FIGURE OFFLINE filename
 A one line title.

inline and offline figures are processed with TRANSPARENT ON, single space, and the current right and left margin. All RIPOFF commands are valid inside an inline or offline figure.

FIGURENUM FLAMBDA (N)

This command sets the current figure number to N. Thus, the next figure will be numbered N. The first figure of a RIPOFF run is by default 1.

FILL LAMBDA (SW)

This command selects whether as many words as possible will be fit on each RIPOFF output line. If FILL's argument is T, then all subsequent output lines will be filled to capacity. Otherwise, the end of an input line will also signal the end of an output line. Initially, the FILL switch is T.

INDENT FLAMBDA (N)

This command sets the current indentation. The single numeric argument gives the number of columns to indent from the left margin. Initially 0. The first print position on a line is whatever the INDENT is plus the left margin value.

ITERATE FLAMBDA [LEFT-MARGIN/,PARAGRAPH-SKIP/,CHAR]

This command starts enumerating items. The arguments are optional. The first argument is a number which specifies how much to indent the numbered paragraphs from the current left margin plus indentation. The second sets the PARAGRAPH-SKIP count. The third argument is the first character to use when iterating. If it is a number then use numbered iteration, otherwise use characters. The next character to use is the next character in the alisp character set i.e. (CHAR (ADD1 (CHAR <last char. used>))). When using numbered iteration it is (ADD1 <last number used>). If not supplied, then the defaults are 0 indent, 1 for PARAGRAPH-SKIP, and numbered iteration starting at 1. The input form for the iterated items is paragraphs, i.e. a blank at the beginning of a line terminates the previous item and begins a new one. For example, numbered reference lists can be generated using this command.

JUSTIFY LAMBDA (SW)

When the JUSTIFY switch is T then each output line will be given a smooth right margin, i.e. there will always be a non-blank character at the right margin. Note, this option is expensive, e.g. in CPU time, and should be considered as something to add for final copies as opposed to drafts. The JUSTIFY switch is initially NIL.

LEFTMARGIN FLAMBDA (N)

This command sets the left margin. It takes one number as an argument. N specifies the number of spaces which will be blank at the beginning of each line. The default is 10, or approximately 1 inch.

LINES-PER-INCH FLAMBDA (N)

Declares N to be the number of lines printed per inch on the printing device. 6 lines per inch is the default. Has to be used when doing sub/super-scripting with lines per inch other than 6.

MALIBU LAMBDA (SW initial-font initial-letter-quality)

This command allows RIPOFF output to be in MALIBU format. If one wants output to the MALIBU printer use the MALIBU command in the first sequence of commands before any text. If sw is T then MALIBU format is used. Default fonts are I and D but RIPOFF will not set the printer values unless values are explicitly given. MALIBU is initially NIL.

- MODEL-PAGE** **FLAMBDA** (WIDTH HEIGHT CHARS-PER-LINE LINES-PER-INCH FIGURE-THRESHOLD%
FIGURE-REDUCTION% TITLE-PAGE-LOST-LINES)
- Specifies printing of "model pages" with column size of WIDTH inches by HEIGHT inches, CHARS-PER-LINE number of characters in a WIDTH inch line, LINES-PER-PAGE number of lines in a column of HEIGHT inches, TITLE-PAGE-LOST-LINES declares the number of lines lost on the first model page (columns 1 and 2) which are reserved for the title of a paper. FIGURE-THRESHOLD% is the maximum percent of a column to be used for figures (a BNUM between 0 and 1, eg .50 is 50%) and FIGURE-REDUCTION% indicates the percent reduction of all figures (they may be too large and may have to be reduced). All arguments are required. Model page format produces output in columns. Columns are numbered and titled just as any page is but the heading lines at the top of each column are not counted as part of the column.
- NULLFACTOR** **LAMBDA** (N)
- If NULLS is on, RIPOFF will print null characters(see NULLS). the number of nulls is defined inside RIPOFF to be (no. specified by RIPOFF)*NULLFACTOR. NULLFACTOR is initially 1, i.e. print exactly what RIPOFF specifies. If the system is very slow (or fast) you may want to print 1/2 (or 2 times) the normal number of nulls. Then, call NULLFACTOR with .5 or 2. any positive number can be used.
- NULLS** **FLAMBDA** (SWITCH)
- This command selects whether null characters will be printed for timing after subscripts, superscripts, overprints, and form feeds. default is ON.
- NEWSECTION** **LAMBDA** (JUMP)
- This function is used to leave spaces between sections and it makes sure that after such space is left out, there is enough space to print the title of the new section and at least one line of the new section. If JUMP is left unspecified it skips two lines.
- OFF** **NIL** ****DATUM****
- An alternate for NIL in functions which turn switches on and off.
eg \$(TRANSPARENT OFF)
- ON** **NIL** ****DATUM****
- An alternate for T in functions which turn switches on and off.
eg \$(JUSTIFY ON)
- PAGE** **LAMBDA** **NIL**
- This command terminates the current page and starts a new page.
- PAGE-LABEL** **FLAMBDA** **LABEL**
- This function takes as argument a label to be used for page numbering and an optional page number, where the numbering is to begin (if left unspecified it defaults to 1).
Eg., \$(PAGE-LABEL A 23) will number pages starting with the number A.23
- PAGE-NUMBER** **LAMBDA** (N)
- This function sets the page number to N.
- PAGENUM** **FLAMBDA** (FLAG)
- This command sets a flag which will determine how page numbering will be done. If the PAGENUM switch is T then page numbering is done one every page except the first. Also, the pages are numbered on the righthand side of the page. If the page numbering switch is NIL then no page numbering will be output. It is intended that other switch values will select other types of page numbering schemes in the future, ex. LEFT-RIGHT would alternate the side of the page the

system will use, etc. Initially the PAGENUM switch is T.

PAGESIZE FLAMBDA (N M)

This command sets the page size, it takes two numbers as arguments. The first is the number of lines per page, the second is the number of columns or print positions per line. The defaults are 66 lines per page and 85 columns per line.

PARAGRAPH LAMBDA NIL

This command terminates the current paragraph and starts a new one.

PARAGRAPH-INDENT FLAMBDA (N)

This command sets the number of spaces which will be used to indent the first line of a paragraph to N. PARAGRAPH-INDENT is initially 5.

PARAGRAPH-SKIP FLAMBDA (N)

This command selects the number of extra lines which will appear between paragraphs to N. Initially, PARAGRAPH-SKIP is 1.

PAUSE LAMBDA (SWITCH)

This command selects whether a pause is to occur at the end of every page. If the value of SWITCH is non-NIL then a control-G (bell) prompt will be issued at the end of each page and processing will not continue until a carriage return is typed at the terminal. If the pause switch is

NIL then RIPOFF continuously lists page after page. This command is useful when preparing "final" versions of a document on noncontinuous form paper (ie, standard bond paper). The switch is initially NIL, no pause. Note the output file must be TTY for PAUSE to have any effect.

PUNCTUATION FLAMBDA PS

This command sets the punctuation symbols after which RIPOFF will print 2 blanks. Initially, PUNCTUATION is (. ; : ? !). One would set PUNCTUATION NIL in reference lists.

REFERENCE FLAMBDA ARGS

A command similar to ITERATE except paragraphs are not numbered. Also, REFERENCE will start at the top of a new page unless NOPAGE is in ARGS. The arguments in order are the left margin to be in force until the END-REFERENCE is encountered and the PARAGRAPH-SKIP. Defaults to the current left margin and a PARAGRAPH-SKIP of 1.

RIGHTMARGIN FLAMBDA (N)

This command sets the right margin. The number N species the last column in which a character will appear.

RIPOFF LAMBDA NIL

RIPOFF is a text system based loosely on a text system named RUNOFF. Briefly, the system allows for simple text formatting of an upper/lower case document containing both text and RIPOFF commands. The commands allow automatic indenting for paragraphs if desired, spacing, margin, and page size settings and many more features. For a complete list of commands see the help option in RIPOFF itself.

The command format of a RIPOFF command is a dollar sign followed by a command. All of the commands are LISP functions which set some flag for the RIPOFF processor or perform some action. The commands which are explicit function calls (eg AUTOPARAGRAPH, TITLE, etc) can appear as single function calls after a dollar sign, or can be put in a

list after a single dollar sign. Two commands which take as arguments lines of text, FIGURE and CENTER, cannot appear in a list of commands.

```
ex -
$(AUTOPARAGRAPH T)
$(FILL NIL)
$((PAGESIZE 67 90)(LEFTMARGIN 10)(RIGHTMARGIN 72))
$CENTER
This line of text.
```

It should be noted that the only place the \$ will be recognized as the "command" macro character is at the beginning of a line.

To start RIPCOM, LOAD the file (RIPCOM CSDLIB) and then call the function RIPOFF of no arguments. You will be prompted to select a mode of operation, eg printing a RIPOFF file, creating a RIPOFF file, etc. Note, that when printing, if the output file is TTY, then RIPOFF output will be sent to the terminal as opposed to a permanent file. To generate formatted output which is in TTY form but on a permanent file use the TTY command.

ai software ltd.
1/11/79 dpm

SECTION	FLAMBDA	DPTH-TITL-NOPAGE
	SECTION:	This function will allow the user to automatically
	-----	number sections, subsections, and sub-subsections.
	*	The call is:
		\$(SECTION n t) or \$(SECTION n t NOPAGE)
		Where n is the DEPTH (1 for a SECTION, 2 for a SUBSECTION,
		and 3 for a SUB-SUBSECTION). This is not the section
		number!
		T is the name for the sub/sub/section. This parameter
		must be a string (eg, be in double quotes).
		if n is 1 and NOPAGE is not present, a new page is started.
		Written by hal
SKIP	LAMBDA	(N)
		This command SKIPs n lines.
SPACING	FLAMBDA	(N)
		This command takes a number as its argument. N specifies the spacing
		which will be in effect until another SPACING command is encountered.
		Default is single spacing, ie 1..
TITLE	LAMBDA	(TITL)
		This command defines a title for the text. The title is a single
		string which is printed at the top lefthand side of each following
		page.
TOP-OF-PAGE	LAMBDA	NIL
		This predicate returns T if the current output line is the first line
		on a page.
TP	LAMBDA	(N)
		This predicate returns T if there are n lines remaining on the
		current page. Purpose is for conditional text formatting which
		is not yet fully developed.
TRANSPARENT	LAMBDA	(SW)
		This command is used to switch to non-formatting mode. It is
		equivalent to (FILL NIL)(AUTOPARAGRAPH NIL) when SW is T.
		If SW is NIL then it is equivalent to (FILL T)(AUTOPARAGRAPH T). Note,
		pagination and headings will still be performed.

TTY LAMBDA (SWITCH)

This command allows RIPOFF output to be in TTY format, ie. no form feeds for line printers will be inserted. If one wants output to the NEC terminal but does not want to wait at the terminal for RIPOFF then use the TTY command in the

^e first sequence of RIPOFF commands before any text. If SWITCH is T, then TTY format will be used. If the output file is specified as TTY then this command will have no effect. TTY is initially NIL.

UNDERLINE FLAMBDA (CH)

Selects CH as the character for underlining. Defaults to _. If CH is not _ then a separate output line is used for underlining.

Below is a table of the character sets available on the MALIBU printer when output is in MALIBU format.

NOTE: font 8 characters are only available if you answer T when asked if you want to use the special characters on the MALIBU.

<u>DECIMAL</u>	<u>ASCII</u> FONT 1	<u>GREEK/MATH</u> FONT 2	<u>APL</u> FONT 3	<u>ITALICS</u> FONT 6	<u>SPECIAL</u> FONT 8
33	!	π	"	!	⌘
34	"	o)	"	⌘
35	#	{	<	#	⌘
36	\$	}	≤	\$	⌘
37	%	}	=	%	⌘
38	&	f	>	&	⌘
39	'	+]	'	⌘
40	([v	(⌘
41)	μ	^)	⌘
42	*	+	*	*	⌘
43	+	+	+	+	
44	,	'	'	'	
45	-	/	+	-	
46	.	~	.	.	
47	/	∆	/	/	
48	0	0	0	0	
49	1	1	1	1	
50	2	2	2	2	
51	3	3	3	3	
52	4	4	4	4	
53	5	5	5	5	
54	6	6	6	6	
55	7	7	7	7	
56	8	8	8	8	
57	9	9	9	9	
58	:	^	(:	
59	;	"	[;	
60	<	π	;	<	
61	=	±	x	=	
62	>	±	:	>	
63	?	f	\	?	
64	@	Σ		@	
65	A	∇	α	A	
66	B	∞	\perp	B	
67	C	Ψ	n	C	
68	D	⊕	L	D	
69	E	+	e	E	
70	F	<	—	F	
71	G	Λ	∇	G	
72	H	Π	Δ	H	

DECIMAL	FONT 1	FONT 2	FONT 3	FONT 6	FONT 8
73	I	↑	l	I	
74	J	>	o	J	
75	K	≠	'	K	
76	L	↓	□	L	
77	M	∅	l	M	
78	N	(τ	N	
79	O)	o	O	
80	P	↓	κ	P	
81	Q	Γ	?	Q	
82	R	θ	p	R	
83	S	Σ	Γ	S	
84	T	+	~	T	
85	U	∑	↓	U	
86	V	⊥	u	V	
87	W	Δ	w	W	
88	X	Ω	∩	X	
89	Y	T	↑	Y	
90	Z	≈	c	Z	
91		√	←		
92	/	Λ	/	/	
93]	≠	+]	
94	^	•	z	^	
95	_	、	-	_	
96	'		◇	'	
97	a	α	A	a	
98	b	β	B	b	
99	c	ψ	C	c	
100	d	φ	D	d	
101	e	e	E	e	
102	f	~	F	f	
103	g	λ	G	g	
104	h	η	H	h	
105	i	ι	I	i	
106	j	≡	J	j	
107	k	κ	K	k	
108	l	ω	L	l	
109	m	{	M	m	
110	n	v	N	n	
111	o	o	O	o	
112	p	p	P	p	
113	q	Υ	Q	q	
114	r	θ	R	r	
115	s	σ	S	s	
116	t	τ	T	t	
117	u	ξ	U	u	

DECIMAL	FONT 1	FONT 2	FONT 3	FONT 6	FONT 8
118	v	x	V	v	
119	w	δ	W	w	
120	x	x	X	x	
121	y	u	Y	y	
122	z	ç	Z	z	
123	{	●	{	{	
124	:]	-	:	
125	}	<	}	}	
126	~	α	\$	~	