NAME

eas – extract an Advice subcircuit from an Advice circuit or library file

SYNOPSIS

eas [**-DV**] [**-s** | **-s=***suffix*] [**-t** *type*] [**-i** *input*] [**-o** *output*] *[name ...]*

DESCRIPTION

Introduction

This program (*eas*) reads standard input and scans it to find the various subcircuits that may be contained in it. The input file is assumed to contain Advice circuit connectivity information usually organized as circuits or subcircuits.

Options

-D This option turns the program debugging more on. This is normally only used during the development of the prgram and is not generally useful otherwise.

-V This option directs the program to output its version and then exit.

-s This option, or this option in the form `-s=`*suffix*, is used to have all circuits that are extracted written out into files with the name of the circuit. Files will always be in lower case characters even if the circuit names are in uppercase. If the form of this option is given with the suffix value, the given suffix will be postpended to the generated file names. The default file name suffix, if not otherwise specified, is *adv*.

-s=*suffix* See the description for the -s option alone.

-t *type* This option restricts the types of subcircuits that are extracted to those that match the specified circuit *type*. Allowable circuit types are:

```
envelope
main
subckt
```

The envelope circuit is the one that does not have a circuit prototype, either `MAIN` or `SUBCKT`. The *main* and *subckt* types correspond to those circuits in the input file that have the prototypes of the given type. Obviously, Advice libraries generally only have circuits of the type *subckt*.

Execution

To extract subcircuits *flip2* and *flip3* from the library file *library.adv* and put them into files by the same names, execute:

```
eas < library.adv flip2 flip3 -s
```

This program will also correctly extract circuits or subcircuits that are defined in a nested way, like in the Pascal language.

EXAMPLES

☞ To extract subcircuits *flip2* and *flip3* from the library file *library.adv* and put them into files by the same names, execute:

```
eas < library.adv flip2 flip3 -s
```

☞ To extract all of the subcircuits from the library file *library.adv* and put them into files by the same names, execute:

```
eas < library.adv -s
```

☞ To extract subcircuits *flip2* and *flip3* from the input file *file.adv* and put them into a common output file named *output.adv*, execute:

```
eas < file.adv flip2 flip3 > output.adv
```

☞ To extract subcircuits *flip2* and *flip3* from the library file *library.adv* that are only of type *subckt* and put them into files by the same names, execute:

```
eas < library.adv flip2 flip3 -s -t subckt
```

☞ To extract just the outer envelope circuit from file *input.adv* and place it in output file *output.adv*, execute:

```
eas < input.adv -t envelope > output.adv
```

☞ To extract just the main circuit from file *input.adv* and place it in output file *output.adv*, execute:

```
eas < input.adv -t main > output.adv
```

ENVIRONMENT VARIABLES

The following environment variables tailor some aspect of the program to the user's preferences.

TMPDIR This environment variable will direct the program to create temporary files in the directory specified. Any specification of a temporary directory that is given on the command line will take precedence over this environment variable.

SEE ALSO

advice(1), runadvice(1), ar(1)

AUTHOR

David A.D. Morano