HACKNET : SPICE Netlist Generator

A netlist generator manual

David Fang

This manual describes the usage and operation of HACKT's `hacknet` spice netlist generator.

This document can also be found online at `http://www.csl.cornell.edu/~fang/hackt/hacknet`.

The main project home page is `http://www.csl.cornell.edu/~fang/hackt/`.

Published by ...

Short Contents

Table of Contents

1 Introduction

`hacknet` is a SPICE netlist generator for the HACKT suite. The input is a HAC file containing production rules, usually connected through instance hierarchy. The output is a hierarchical netlist with devices, subcircuits, and instances. The output should be suitable for SPICE-like circuit simulators.

How it works...

2 Usage

This chapter describes `hacknet`'s command-line options, and configuration file options.

Usage: '`hacknet [`*`options`*`]` *`obj-file`*'

The resulting netlist is printed to '`standard-out`', so it is common practice to redirect it to a file. Diagnostic messages will appear in '`standard-error`'.

2.1 Option Summary

For options that take an argument, the space between the flag and the argument is optional.

`-c` *file* [User Option]

Parse configuration options from *file*. Options are of the form `key=values` with no space characters separating the `=`. Values may be singleton, comma-separated, or omitted. The same options can also be passed in through the command-line via '`-f`'. This option is repeatable and cumulative. See Section 2.2 [Configuration Options], page 3.

`-f` *options...* [User Option]

Parse configuration options from *options*. Options are of the same key-value format as in the configuration file, see option '`-c`'. Options are space-separated instead of newline-separated. This option is repeatable and cumulative. See Section 2.2 [Configuration Options], page 3.

`-h` [User Option]

Help. Print usage and exit.

`-H` [User Option]

Describe all configuration options. See Section 2.2 [Configuration Options], page 3. See also the installed documentation for '`hacknet.info,html,pdf`'.

`-t` *type* [User Option]

Instead of using the top-level instances in the source file, instantiate one instance of the named *type*, propagating its ports as top-level globals. In other words, use the referenced type as the top-level scope, ignoring the source's top-level instances. Convenient takes place of copy-propagating a single instance's ports.

2.2 Configuration Options

There are two ways to pass configuration options to `hacknet`. One is through the '`-f`' option on the command-line, the other way is to pass them in through a configuration file with the '`-c`' option. The option value specifications share the same syntax: *key=values* where *values* can be blank, a single value, or a comma-separated list of values. A key-value specifier is not permitted to have spaces in the string!

In a configuration file, blank lines are ignored, as well as lines that begin with `#` (pound).

`lambda` [User Option]

Technology-dependent scaling factor for device lengths and widths, the multiplier factor applied to lengths and widths specified in PRS. Default: 1.0

`length_unit` [User Option]

Suffix-string to append to emitted length and width parameters. Can be a unit such as "u" or "n", or exponent such as "e-6" or "e-9". Default: u (micron)

`std_n_width` [User Option]

Default width (in lambda) for NFETs used in logic, where unspecified.

`std_p_width` [User Option]

Default width (in lambda) for PFETs used in logic, where unspecified.

`std_n_length` [User Option]

Default length (in lambda) for NFETs used in logic, where unspecified.

`std_p_length` [User Option]

Default length (in lambda) for PFETs used in logic, where unspecified.

`stat_n_width` [User Option]

Default width (in lambda) for NFETs used in keepers (staticizers), where unspecified.

`stat_p_width` [User Option]

Default width (in lambda) for PFETs used in keepers (staticizers), where unspecified.

`stat_n_length` [User Option]

Default length (in lambda) for NFETs used in keepers (staticizers), where unspecified.

`stat_p_length` [User Option]

Default length (in lambda) for PFETs used in keepers (staticizers), where unspecified.

`nested_subcircuits` [User Option]

If this option is set to 1, then emit local subcircuits as nested definitions within their used definitions. Not sure which variants of SPICE support this. Default: 0

`emit_top` [User Option]

If set to 1, include the top-level instances in the netlist output. Setting this to 0 is useful for producing a library of subcircuit definitions for every type that was instantiated, recursively w.r.t dependencies. Default: 1

Concept Index