NAME

ccif - convert CIF to layout database

SYNOPSIS

ccif [-4bforstvxz] [-u unit] [-w size] inputfile

OPTIONS

-4 Test non-orthogonal input conform 45 degree.

-b Sound the bell if the program is finished.

-f Enables cell overwrite (force) mode.

-o Use no-origin mode (default).

-r Give no rounding messages.

-s Do only a syntax check.

-t Generate a terminal point or terminal box (depending on the value of the option -w). For CIF files, where the terminal specification is not coincide with the previous box specification. This option is for the old 94 user code interpretation (option -z).

-v Enable the verbose mode.

-u Set the cifunit of the parameters in the CIF file, the default cifunit is 100.

-w Specifies the size of terminal boxes that are generated because of the use of the option -t (default 0).

-x Use origin mode.

-z Use the user 94 codes for terminals (old mode).

DESCRIPTION

Ccif adds the layout description of one or more cells (symbols in CIF) to the ICD database. If a cell description is read, *ccif* creates a new cell directory, and translates the CIF cell description into an internal database format.

The cell descriptions are read from the specified CIF file, which must be the last argument. In most cases a file with suffix ".cif". The input must be conform the syntax rules of CIF. If not, a syntax error will be the result, and an appropriate error message will be generated.

IMPORTANT: *ccif* requires that cells (symbols) that are called in the CIF file are defined earlier in the file. NEW: *ccif* can use the comment after a 'C0'-call to set the cell name, if symbol0 is not earlier defined in the file. Thus, you can use library cells not defined in the CIF file.

To help you in making a correct cell definition you can invoke *ccif* with option -s. In that case *ccif* performs only a syntax check, and two kinds of semantic checks, i.e.:

1) a cell may not be redefined,

2) a cell called in a cell description must already exist.

With the origin mode (option -x) the lower-left corner of the bounding box of a sub-cell is used as the translation vector reference point. Note that, if the bounding box information of sub-cells in the database is incorrect, this sub-cells can be incorrectly placed.

If an error is detected the cell will not be added to the database. This can result in subsequent error messages, complaining about the fact that the cell does not exist.

The unit of the parameters is assumed to be in centi microns. Since the unit of the database is lambda, each parameter is multiplied by a resolution value:

new-parameter = parameter * resolution

resolution = (A/B) ∗ (1/cifunit) ∗ (1/projectlambda)

where A and B are the scaling factors of the Definition Start statement, and cifunit is 100 (default). Thus, with a cifunit of 100 and a projectlambda of 0.01 micron the scaling (resolution) used is A/B.

AUTHOR

J. Annevelink, S. de Graaf, T.G.R. van Leuken.

FILES

`NELSISPROJECT/.dmrc`

`NELSISPROJECT/layout/impcelllist`

`NELSISPROJECT/layout/celllist`

For each *cell* in the celllist:

`NELSISPROJECT/layout/`*cell*`/annotations`

`NELSISPROJECT/layout/`*cell*`/box`

`NELSISPROJECT/layout/`*cell*`/info`

`NELSISPROJECT/layout/`*cell*`/mc`

`NELSISPROJECT/layout/`*cell*`/nor`

`NELSISPROJECT/layout/`*cell*`/term`

LIMITATIONS

The current working directory must be the project directory.

SEE ALSO

addproj(1ICD), impcell(1ICD), mkpr(1ICD), xcif(1ICD).

DIAGNOSTICS

All error messages are intended to be self-explanatory.