NAME

cedif - compile an EDIF source into the Nelsis database

SYNOPSIS

cedif [options] [edif_source_file]

OPTIONS

-N Set the target language to "Nelsis" (default).

-S Set the target language to "Seadif", see also esea(1SDF).

-P Set the target language to "Pseudo-Seadif".

-b <lib>

Translate the (explicit) EDIF library named <lib>. The default <lib> is the library with the same name as the current working directory (sans leading path name).

-B <lib>

Translate the implicit EDIF library named <lib>. An implicit library is a library that is not defined in the EDIF source, but only referenced through the EDIF libraryRef statement. See also the section "Explicit and implicit libraries" in this manual page.

-o <outputfile>

Output goes to <outputfile> (this option is a no-op if the target language is "Nelsis").

-m <lib1>,<lib2>

If the EDIF source references the external library <lib1> then replace this with a reference to the external library <lib2>. Multiple **-m** mapping options are allowed.

-l Only list the (explicit) libraries that appear in the EDIF source. Do not translate.

-L Only list the implicit libraries that appear in the EDIF source. Do not translate. See also the option **-B <lib>**.

-E If the EDIF source contains "external" declarations of external cells, accept references to these cells without checking that they really exist. This is of course a bad thing, but sometimes it may be an acceptable way to deal with messy EDIF sources.

-h Help, print a list of options.

DESCRIPTION

Cedif compiles an EDIF version 2.0.0 source file into the Nelsis database.

The EDIF language contains the notion of a *library* to group cells. The program *cedif* relates such an EDIF library to a Nelsis *project* and vice versa. To stress this relation, the name of a Nelsis project should match the name of the corresponding EDIF library. For example, if we want to translate the EDIF library "newlib", we create a Nelsis project called "newlib" and call *cedif* from inside this project directory. Suppose that an EDIF cell in "newlib" references a cell "nand" in the EDIF library "cmoslib" then *cedif* translates this into a Nelsis *model call* to the cell "nand" in project "cmoslib".

You can convince *cedif* to deviate from this behavior by using the options "**-b <lib>**" and "**-m <lib1>,<lib2>**". For example, the option "**-b otherlib**" would compile the EDIF library "otherlib" into the current project "newlib", even though the names do not match.
And the option "**-m cmoslib,oplib**" would generate a model call to cell "nand" in project "oplib" whenever the EDIF source references a cell "nand" in library "cmoslib".

Note that you must have imported all cells that the EDIF library references before attempting to translate the EDIF library. *Cedif* complains and exits if it cannot resolve a reference to an imported cell. See also the Nelsis commands *addproj*(1ICD) and *impcell*(1ICD).

EXPLICIT AND IMPLICIT LIBRARIES

Normally we only deal with libraries that are explicitly defined in the EDIF source by means of the EDIF "library" statement. If such an explicitly defined library references cells in libraries that are not defined by the EDIF source, then this can be interpreted as an implicit definition of these undefined libraries. For example, if cell "des" in library "newlib" references the cell "nand" in library "cmoslib", this can be regarded as an implicit definition of "nand" and "cmoslib". We can carry this even further: if a net in cell "des" references port "F" of "nand", then we can regard this as an implicit interface definition of "nand", that is: we assume that "nand" has a port named "F".

In this way, it is possible to compile an implicitly defined library into the Nelsis database.
The options "**-B <lib>**" and **-L** support implicit libraries. One example of their usage is when we want to translate the EDIF library "newlib" but we do not actually have "cmoslib" hanging around. In this case we first translate the implicit library "cmoslib" to Nelsis by creating the project "cmoslib" and then typing "cedif -B cmoslib file.edif". Following this, we create the project "newlib" and import all circuit cells from "cmoslib" into "newlib". Finally we can translate newlib by executing "cedif file.edif" from within the project "newlib".

AUTHOR

Paul Stravers, Patrick Groeneveld.

SEE ALSO

addproj(1ICD), esea(1SDF), impcell(1ICD), xedif(1ICD).