

## NAME

`xsls` - extract SLS network description out of the database

## SYNOPSIS

`xsls` [-defhirtv -w width -C file -D name -F outfile -X lib] cell

## OPTIONS

- d** Use original database names for instance names.
- e** Expand name arrays into single names.
- f** Send output to a file *cell.sls* instead of to *stdout*.
- h** Also extract hierarchy (used local sub-cells).
- i** Also extract imported sub-cells (when option **-h** is used).
- r** Do not generate extern network definitions for sub-cells that are not retrieved.
- t** Do not output unconnected instances.
- v** Enable verbose mode.
- w** Use *width* for the output listing.
- C file** Use the specified *file* as the control file.
- D name**  
Define label *name* for conditional reading of the control file.
- F outfile**  
Send output to the specified *file*.
- X lib** Exclude specific certain library cells from the listing (when options **-hi** are used). This is very useful if you have a design with your own imported libraries. This option may be given several times. There are two exclude possibilities: (1) all lib's starting with absolute path (for example "-X /usr/ocean"); (2) the lib with basename (for example "-X primitives").

## DESCRIPTION

*Xsls* is a program to extract an SLS circuit description out of the database. Default, only a circuit description of the cell itself is extracted and no sub-cells are extracted. When using the **-h** option the program will also extract all (local) sub-cells that are not a function (see "SLS: Switch-Level Simulator User's Manual") or a device.

Whether or not a cell is a device is determined by (1) the use of the program *xcontrol(IICD)* for new projects or *device(IICD)* for old projects and/or *putdevmod(IICD)* (see the manual pages of these programs), or by (2) a set of built-in device names: "res" for resistor, "cap" for capacitor, "nenh" for an n-enhancement transistor, "penh" for a p-enhancement transistor, "ndep" for an n-depletion transistor, "nnp" and "pnp" for bipolar transistors and "d" for a diode.

## THE CONTROL FILE

For *xsls* a control file may be used. The default name of the control file is "xslsrc". First, the program tries to read this file from the current working directory. Next, it tries to open the control file in the process directory. Note, the control file may also be used for other programs.

The following specification is useful with the **-i** option:

exclude\_project *path\_name*

to exclude specific certain imported projects from the net-listing (see the **-X** option).

## CONTROL FILE CONDITIONALS

It is possible to use "#if", "#else" and "#endif" statements, to make the reading of the control file depended of the invoking program. To be seen, the '#' character must be the first character on a line. This

conditional statements may not be nested like the C-preprocessor. Normally, from the top of the file reading is set to true. A "#else" statement changes the state from true to false or false to true. A "#endif" statement changes the state always to true. A "#if" statement changes only the state to true if one of the specified conditions is true for the invoking program (else to false). Note that you can also define one condition on the command line with the **-D** option.

The following condition is true for *xsls*:

```
#if SLS
```

#### **EXAMPLES**

```
% xsls -h latch
```

#### **AUTHOR**

A.J. van Genderen

#### **MODIFIED BY**

S. de Graaf

#### **FILES**

*cell.sls* (output file, when option **-f** is used)

*xslsrc* (default control file for xsls/xvhdl/...)

#### **SEE ALSO**

*csls*(1ICD), *putdevmod*(1ICD), *space*(1ICD), *xcontrol*(1ICD), *xedif*(1ICD), *xnle*(1ICD), *xspice*(1ICD), *xvhdl*(1ICD).