NAME

orcad_convert – convert OrCAD L.P. SDT III/IV source libraries and schematics to KiCad / gEDA

SYNOPSIS

orcad_convert [*options*] [*input_file*] [*output_file*]

orcad_lib_convert [*options*] [*input_file*] [[*output_file*]]

DESCRIPTION

orcad_convert will dump the contents of an OrCAD L.P. SDT III/IV schematic (.sch) *input_file* or will convert it to a KiCad eeschema or gEDA gschem text-based schematic format in *output_file*. If *input_file* is not specified, standard input will be read. If *output_file* is not specified, standard output is used.

orcad_lib_convert will dump the contents of an OrCAD L.P. SDT III/IV library source (.src) *input_file* or will convert it to a KiCad component library (.lib) *output_file* or to gEDA symbol files (.sym) in directories having the same name as the input library file. In this last case, *output_file* is not specified. If *input_file* is not specified, standard input will be read. If *output_file* is not specified, standard output is used (KiCad and listing only).

orcad_convert OPTIONS

(no option)
: outputs interpreted schematic file information

–h displays program and option information

–s input file statistics; activated with **-V**

–g gEDA (gschem) output format

–k KiCad (eeschema) output format

–m module ports connected on both sides (KiCad output only, see MODULE PORTS)

–n module ports translated into named nets (KiCad and gEDA output)

–V additional debug information is output

orcad_lib_convert OPTIONS

(no option)
: outputs a parsed version of the input file

–h displays program and option information

–g gEDA symbol files (.sym) output in per-library directories under the same name

–k KiCad component library (.lib) output

–p power pins are visible (default is invisible)

–s file statistics

–V additional debug information is output; if -V (verbose) is used along with the -g (gEDA) or -k (KiCad) output options, debug information will be included in the output file(s) as comments.

HISTORY AND INTENDED USE

orcad_convert was written to translate OrCAD L.P. SDT III/IV schematic files into a current-day open format (KiCad, gEDA), mainly for archiving purposes. This also provided a good opportunity to evaluate the capabilities of the two schematic capture programs.

The companion **orcad_lib_convert** program is required to render the stock or custom component libraries,

which should be in source (.src) format, into corresponding KiCad libraries (.lib) or gEDA symbol (.sym) files grouped by library directory. Please follow the workflow examples, using specific file extensions, in order to maintain sanity.

Of this exercise, KiCad output appears to be preferable over the gEDA one, although not without its caveats.

If you are interested in the program's processing details and decisions, please refer to the extensive introductory comments in the program source files.

February 2020 - started as a lockdown project :D

May 2020 - version 2.9 supporting both KiCad and gEDA output

KiCad WORKFLOW EXAMPLE

`./orcad_lib_convert -k TTL.SRC TTL.lib`

Translates the original library TTL.SRC to a KiCad library, named TTL.lib, also producing TTL.SRC.txt with component information (PARTNAME X_SIZE Y_SIZE LIBRARY_NAME) which is used for correct placement of a component when its orientation is not the default.

`source orcad_lib_convert_kicad.sh`

An **example** script for converting libraries is distributed with the original OrCAD L.P. application. Following the individual library conversions, as above, the script also includes a concatenation part aimed at dictating the library order; this is to determine which part will be used over others under the same name. The resulting sequence of parts is placed in a file called **orcad_parts.txt**. This file is used by the **orcad_convert** program to determine the proper placement of a component, if its orientation is not the default. Modify this script, as you see fit, or create your own.

Partial content of **orcad_lib_convert_kicad.sh** follows:

```
 ./orcad_lib_convert -k ALTERA_P.SRC ALTERA_P.lib
 ./orcad_lib_convert -k CMOS.SRC CMOS.lib
 ./orcad_lib_convert -k TTL.SRC TTL.lib
 ./orcad_lib_convert -k DEVICE.SRC DEVICE.lib
cat ALTERA.SRC.txt.br
    CMOS.SRC.txt.br
    DEVICE.SRC.txt.br
    TTL.SRC.txt.br
    > orcad_parts.txt
echo Total parts:
wc orcad_parts.txt
```

`./orcad_convert -k -n halfadd.sch halfadd.k.sch`

Convert an OrCAD schematic to KiCad format, also translating module ports to nodes (nets). As the schematic files of all three applications under consideration end in **.sch**, the **.k.sch** extension will remind us that this is a schematic in KiCad (eeschema) format.

If a schematic component name is found in the **orcad_parts.txt** file, the corresponding library name is prepended to the component name, e.g. TTL:74HC94

The resulting schematic may now be opened with KiCad. Make sure you define the libraries used in each project. Library names in **orcad_parts.txt** should match the "Nickname" in KiCad's "Manage Symbol Libraries" window; the respective information is entered in the **sym-lib-table**

file present in each KiCad project directory. e.g.
`((lib (name TTL)(type Legacy)(uri ...) ...&)`

OrCAD SDT power components (arrow, bar, circle, wave) and markers (stimulus, vector, trace and layout directives) are not library parts; a special library called **pwr.lib** was created to compensate and this must be included in the KiCad library definitions.

gEDA WORKFLOW EXAMPLE

./orcad_lib_convert -g TTL.SRC

Translates the original library TTL.SRC to symbol (.sym) files, under a directory named TTL, also producing a file with component information (PARTNAME X_SIZE Y_SIZE LIBRARY_NAME) which is used for correct placement of a component when its orientation is not the default.

Note that gEDA does not support component ALIASES. A symbolic link with the component ALIAS name is created. A drawback is that when this component symbol is used in a **new** drawing, it will have the partname of the 'master' component.

source orcad_lib_convert_geda.sh

An **example** script for converting libraries distributed with the original OrCAD L.P. application. Following the individual library conversions, as above, the script also includes a concatenation part aimed at dictating the library order; this is to determine which part will be used over others under the same name. The resulting sequence of parts is placed in a file called **orcad_parts.txt**. This file is used by the **orcad_convert** program to determine the proper placement of a component, if its orientation is not the default.
Modify this script as you see fit, or create your own.

Partial content of orcad_lib_convert_distribution.sh follows:

```
 ./orcad_lib_convert -g ALTERA_P.SRC
 ./orcad_lib_convert -g CMOS.SRC
 ./orcad_lib_convert -g TTL.SRC
 ./orcad_lib_convert -g -p DEVICE.SRC
cat ALTERA.SRC.txt.br
    CMOS.SRC.txt.br
    DEVICE.SRC.txt.br
    TTL.SRC.txt.br
    > orcad_parts.txt
echo Total parts:
wc orcad_parts.txt
```

Note the **-p** option which tells the program to render the power (PWR) pins invisible for better schematic readability.

./orcad_convert -g halfadd.sch halfadd.g.sch

Convert an OrCAD schematic to gEDA format. As the schematic files of all three applications under consideration end in **.sch**, the **.g.sch** extension will remind us that this is a schematic in gEDA (gschem) format.

The resulting schematic may now be opened with gEDA. A file in the user's home directory (**~/.gEDA/gafrc**) should include the following lines (replace quoted content with actual paths):

```
(component-library "path_to_an_individual_component_library")
(component-library-search
```

```
"path_to_a_directory_with_symbol_directories")
(source-library "path_to_the_working_directory")
```

The third line is used to indicate where hierarchical drawing "source" files - namely the files lower in the hierarchy - may be found.

OrCAD SDT power components (arrow, bar, circle, wave) and markers (stimulus, vector, trace and layout directives) are not library parts; a special library directory called **PWR** with corresponding symbol files (.sym) was created to compensate. The PWR directory should be placed along with the other library directories in the file hierarchy.

POST-PROCESSING

Following conversion to the preferred format, the following items should be checked:

KiCad **Module ports** - depending on the type and orientation of the module port, these may be left unconnected. Options **-m**, trying to connect both sides of a module, and **-n**, converting the module port into a named node (net), are possible workarounds.

Hierarchical drawings - the sheet filename for schematic blocks in higher-level schematics may need changing (e.g. from halfadder.sch to halfadder.k.sch)

Bus entries (bus rippers) - Orcad SDT does not have a bus-to-bus entry; all bus entries are represented in the converted schematics by wire-to-bus entries; bus-to-bus segments should be manually corrected.

Component libraries - The library for a component may be redefined by manual or semi-automatic substitution at the KiCad drawing level. An example script called **add_libraries.sh** is provided.

gEDA **Grid** - Set the snap grid spacing to 10 mils (Options > Snap Grid Spacing > 10. gschem expects pins to be on a 100 mil grid.

Nets - A "wire" passing at the connecting edge of a component pin **will be automatically connected to the pin**. gEDA does not support explicit junctions. Obviously this is wrong but I can't think of an easy workaround. **CHECK YOUR OUTPUT**.

Module ports - module ports are represented as embedded components with connections (pins) on both sides. Option **-n** for converting the module port into a named node (net) is also possible. Note that gEDA does not support bus module ports and such module ports are not shown connected in the resulting schematic.

Hierarchical drawings - the sheet filename for schematic blocks in higher-level schematics may need changing (e.g. from halfadder.sch to halfadder.g.sch)

Bus entries (bus rippers) - Orcad SDT does not have a bus-to-bus entry; all bus entries are represented in the converted schematics by wire-to-bus entries; bus-to-bus segments should be manually corrected.

Symbol editing - The top left corner of a component symbol translated using **orcad_lib_convert** is used as the anchor point, for compatibility with the OrCAD schematics being translated. When opening a symbol with **gschem** or through the Hierarchy > Down Symbol command only the top of the symbol is shown. To access the full symbol, use the command **Edit > Symbol Translate > 3000**. Also, note that gEDA (gschem) will rotate a selected component using the cursor coordinates as the rotation center.

Symbol size - The symbols created from OrCAD component libraries maintain their original size.

Stock gEDA symbols appear to be larger in size. If you want to use OrCAD component drawings in new designs, consider increasing the SCALE parameter in **orcad_lib_convert.h** from 100 to 200.

FURTHER DEVELOPMENT

Not expected, unless someone finds a glaring bug which cannot be self-corrected :D

If something is amiss in the provided material, please drop me a line and I will correct as soon as possible.

REFERENCES (updated 2-Jan-2025)

- orcad_schematic_file_format (own information for SDT III/IV, also checked against SDT IV information from https://www.pcengines.ch/orcadff.htm) -- consult the source files or the default diagnostic output for differences / additions.
- KiCad File formats (https://dev-docs.kicad.org/en/file-formats/index.html)
- gEDA/gaf File Format Document (http://wiki.geda-project.org/)
- gEDA/gaf Master Attribute Document
- gEDA_netattrib.pdf
- gEDA Hierarchical schematics

COPYRIGHT