

NAME

rtl2gds – VLSI Back End Design utility to convert rtl to gds2

SYNOPSIS

rtl2gds [*run options*] [*input files/settings*] [*library files/settings*] [*timing constraints*]

DESCRIPTION

rtl2gds utility takes a single rtl file as input and completes the back end design flow including power calculation. Input to the utility is RTL file and design constraints. Output is GDS2 (mask layout). rtl2gds is a wrapper utility around SoC Encounter, Design Compiler and ModelSim to accomplish the back-end design flow.

RTL2GDSHOME env variable pointing to the installation directory has to be set prior to running. All default paths are relative to \$RTL2GDSHOME directory.

If RTL2GDSHOME is not set default tool installation directory is taken as /usr/bin. In a similar way default library directory is taken as /usr/bin/rtl2gds_install/LIB.

OPTIONS

run options

-genScr=<TOP-DIR>

Generate only the scripts. No tool is run. Creates the top directory structure with relevant sub-directories and scripts under the hierarchy "TOP-DIR". This is the first step in back-end design flow. Subsequently individual steps are invoked using -syn/-pnr/-sim/-all options

-syn run the synthesis alone.

-pnr run Placement and Route alone.

-sim run simulation and generate VCD file.

-pow run VCD based power calculation alone.

-all run synthesis, pnr, simulation and power calculation in the order.

-help print a help message and exit.

Input files/settings options

-rtl=<rtl-filename/rtl-filelist>

rtl file to be synthesised. RTL source files can be specified as a single file or multiple files in a list. rtl2gds utility identifies the language (verilog/vhdl) from the file extension. Valid extensions are .vhd/.vhdl for vhdl and .v for verilog. Any other extensions is taken as rtl file list. (currently supported only for vhdl file list)

When used as -rtl=PATH/sample.v, rtl source is a single verilog file.

When used as -rtl=PATH/sample.vhd or -rtl=PATH/sample.vhdl, rtl source is taken as a single vhdl file.

When used as -rtl=PATH/vhdlfile.list, rtl source is taken as the files in the vhdlfile.list.

Sample vhdlfile.list

```
/home/users/rtl/dataPath.vhd myLib1
```

```

/home/users/rtl/controlPath.vhd myLib2
/home/users/rtl/system.vhd work
/home/users/rtl/top.vhd

```

In the example vhdfile.list, dataPath.vhd will be compiled to library myLib1, controlPath will be compiled to library myLib2 and system.vhd will be compiled to work library. top.vhd will be compiled to work library as the default library is work.

If no library is specified, the rtl2gds utility assumes the file to be compiled into work library. All lines should be pointing to a rtl file and no comments allowed. Order of the rtl source files is users responsibility. If not provided the proper order, synthesis may fail.

Sample vhdfile.list is provided in TOP-DIR/rtl/vhdl.list

-rtl_top=<rtl top-design-unit name>
top design entity in the rtl.

-tb=<test bench filename>
test bench file that has "rtl_top" instantiated as DUT (Design Under Test)

-tb_top=<test bench top unit name>
test bench top entity

NOTE:

Provide full path for files/directories. Provide RTL2GDShome env variable pointing to the installation directory

Library files/settings options

-lib=<location of LIB directory>
variable pointing to the LIB directory

-tech=<target technology>
target technology to be employed

--lib_db=<location of DB library file>
Library needed by synthesis tool.

--lib_lef=<location of LEF library file>
Library needed by pnr tool. (Physical library)

--lib_tlf=<location of Timing library file>
Library needed by pnr tool. (Timing library). Can provide .tlf or .lib(liberty) formats

--lib_v=<location of verilog library file>
Library needed by simulation tool. (Functional library)

NOTE:

Provide full path for files/directories. Provide RTL2GDShome env variable pointing to the installation directory. --lib_* options take precedence over LIB env variable and -lib option.

Timing Constraints options

-frequency=<value> (in MHz)
target frequency of the design in MHz.

-max_area=<value> (in um2)
target area value

-io_input_delay=<value> (*in ns*)
io input delay value

-io_output_delay=<value> (*in ns*)
io output delay value

-clk_latency=<value> (*in ns*)
clock latency value. This value is used as clock_insertion_delay target in Clock-Tree-Synthesis

-clk_uncertainty=<value> (*in ns*)
clock uncertainty value. This value is used as clock_skew target in Clock-Tree-Synthesis

USAGE

Typical usage of the utility after installation is explained below.

Set the RTL2GDSHOME env-variable to the directory of installation. The first step in backend-design flow is the creation of relevant directories and subdirectories so that the inputs/results are organized properly.

rtl2gds -genScr=TOP-DIR

"TOP-DIR" can be any directory under which the flow is organized. This command will not run any tools; only creates the directory structure with relevant scripts and templates.

examples:

```
rtl2gds -genScr=/home/users/user1/counter_design
```

Next step in the flow is synthesis.

The tool needs to be executed from "TOP-DIR" for synthesis. Only a single rtl file is supported. (See LIMITATIONS section for further information). RTL files may be simply concatenated, but the order should be maintained. Provide relevant options. Minimum options required are **-syn**, **-rtl**, **-rtl_top**

rtl2gds -syn -rtl=<rtl-file> -rtl_top=<rtl-top-design-unit>

examples:

```
# cd TOP-DIR
```

```
# rtl2gds -syn -rtl=/home/user/rtl/counter.vhd -rtl_top=counter -lib=/usr/local/LIB -tech=tsmc018
```

```
# rtl2gds -syn -rtl=/home/user/rtl/vhdl.list -rtl_top=counter -lib=/usr/local/LIB -tech=tsmc018
```

```
# rtl2gds -syn -rtl=/home/user/rtl/counter.vhd -rtl_top=counter --lib_lef=/cad/LIB/tsmc180.lef --lib_tlf=/cad/LIB/tsmc180.tlf
```

```
# rtl2gds -syn -rtl=/home/user/rtl/counter.vhd -rtl_top=counter -frequency=100
```

Further steps in the flow can be executed similar to synthesis. For simulation minimum options needed are **-sim**, **-tb**, **-tb_top**

RESULTS/DIRECTORIES

Individual readMe's are provided in the respective directories for a detailed explanation. TOP-DIR/Back-End_DesignFlow.pdf provides an explanation of the different steps in back-end design flow.

TOP-DIR/synthesis

The directory structure is organised as follows.

1. logs: logs of the synthesis runs are redirected here.

2. op_data: Results of the synthesis runs are saved here.
3. reports: Reports of the synthesis runs are redirected here. Reports generated are design reports, "TOP-DIR/synthesis/reports/*design*.rpt" at various stages of the run. timing reports, "TOP-DIR/synthesis/reports/*timing*.rpt", final timing information. area, cell, power and qor reports.
4. run: run scripts/other executables.
5. scripts: Directory for all tcl/perl scripts.
6. tmp: temporary directory to save intermediate results, calculations etc.

NOTE:

1. "TOP-DIR/synthesis/scripts/compile_dc.tcl": Main tcl script called by dc_shell
2. "TOP-DIR/synthesis/scripts/technology.tcl": Contains all the technology settings. Provide proper values using -tech and -lib options. Technology library settings may need to be modified if the library hierarchy is different. Editing to be done in "TOP-DIR/template/technology.tcl"
3. "TOP-DIR/synthesis/scripts/rtl.list": This file points to the rtl files. Provide proper values using -rtl and -rtl_top options.
4. "TOP-DIR/synthesis/scripts/constraints.tcl": This file contains, the design constraints commands. Provide proper values using -frequency and other timing-constraints options.
5. "TOP-DIR/synthesis/run/run_dc.bash": Run script.

TOP-DIR/pnr

1. logs: logs of the pnr runs are redirected here.
2. op_data: Results of the pnr runs are saved here.
3. run: run scripts/other executables.
4. scripts: Directory for all tcl/perl scripts.
5. tmp: temporary directory to save intermediate results, calculations etc.
6. ip_data: extra input files.
7. conf: Encounter initial configuration script is kept in this folder.
8. reports: Reports of the pnr runs are redirected here.

Reports generated are design/timing reports, CTS reports, gateCount reports, summary report, verify report

NOTE:

1. "TOP-DIR/pnr/scripts/pnr.tcl" is the main tcl script called by encounter. Inside "TOP-DIR/pnr/scripts/pnr.tcl" The conf points to netlist, timing constraints, technology settings (.tlf, .lef) Technology library settings may need to be modified if the library hierarchy is different. Editing to be done in "TOP-DIR/template/encounter.conf"
2. Edit "TOP-DIR/template/pnr.tcl" to change all the default options like floorplan aspect ratio, power-stripe width etc.
3. Run script is "TOP-DIR/pnr/run/run_pnr.bash"
4. By default all the reports are generated in "TOP-DIR/pnr/reports/" directory.

TOP-DIR/simulation

1. tb: The testbench code may be kept in this directory.
2. vcd: vcd dump is redirected here.
3. run: ModelSim tool run directory.

NOTE:

1. "TOP-DIR/simulation/run/simulate.do" is the main script called in modelsim. Technology library settings may need to be modified if the library hierarchy is different. Editing to be done in "TOP-DIR/template/simulate.do"
2. "TOP-DIR/simulation/run/run_sim.bash" is the run script

TOP-DIR/pnr (*Power calculation based on VCD using Encounter*)

This is the same directory used for place-n-route

NOTE:

1. "TOP-DIR/pnr/scripts/power.tcl" is the main tcl script called by encounter.
2. Run script is "TOP-DIR/pnr/run/run_power.bash"
3. All reports are generated in "TOP-DIR/pnr/reports/" directory.

TOP-DIR/rtl

Directory is available for proper organization of the rtl-code. User may link/copy all the rtl files in TOP-DIR/rtl/code and use "-rtl" option

TOP-DIR/tmp

For temporary purpose.

LIMITATIONS

1. Only a single rtl file is supported. RTL files may be simply concatenated, but the order should be maintained.
2. For VHDL the following port types are only supported: bit, bitvector, stdlogic, stdlogicvector
3. Clock pin in rtl needs to be "clk"
4. Works only for single clock designs. (And one external virtual clock)
5. Only single corner STA/optimization supported. (min-max libraries cannot be provided in single run)
6. "TOP-DIR/pnr/scripts/gds2_encounter.map" is for OSU library.
If a different foundry is targeted, this file need to be changed manually.
7. Applicable to *Cadence SoC Encounter version-8.1 or version-6.2*, *Synposys Design Compiler version-2006.06*, *Mentor ModelSim version SE-64 6.2*
8. By default dft/scan insertion is not done at synthesis stage. Change the variable "enable_scan" in TOP-DIR/template/compile_dc.tcl to 1, to enable scan.

SEE ALSO

Back-End_DesignFlow.pdf in the TOP-DIR.
Individual readMe's in the pnr/simulation/synthesis/rtl directories.

AUTHOR

Arun. C (arunc@ee.iitb.ac.in)

The rtl2gds project is done under the supervision of Prof. Madhav. P. Desai (madhav@ee.iitb.ac.in) at IIT-Bombay.

COPYRIGHT

GPL IIT-Bombay