OpenROAD

OpenROAD Team

Nov 28, 2023

CONTENTS

The OpenROAD ("Foundations and Realization of Open, Accessible Design") project was launched in June 2018 within the DARPA IDEA program. OpenROAD aims to bring down the barriers of cost, expertise and unpredictability that currently block designers' access to hardware implementation in advanced technologies. The project team (Qualcomm, Arm and multiple universities and partners, led by UC San Diego) is developing a fully autonomous, open-source tool chain for digital SoC layout generation, focusing on the RTL-to-GDSII phase of system-on-chip design. Thus, OpenROAD holistically attacks the multiple facets of today's design cost crisis: engineering resources, design tool licenses, project schedule, and risk.

The IDEA program targets no-human-in-loop (NHIL) design, with 24-hour turnaround time and zero loss of power-performance-area (PPA) design quality.

The NHIL target requires tools to adapt and auto-tune successfully to flow completion, without (or, with minimal) human intervention. Machine intelligence augments human expertise through efficient modeling and prediction of flow and optimization outcomes throughout the synthesis, placement and routing process. This is complemented by development of metrics and machine learning infrastructure.

The 24-hour runtime target implies that problems must be strategically decomposed throughout the design process, with clustered and partitioned subproblems being solved and recomposed through intelligent distribution and management of computational resources. This ensures that the NHIL design optimization is performed within its available `[threads * hours]` "box" of resources. Decomposition that enables parallel and distributed search over cloud resources incurs a quality-of-results loss, but this is subsequently recovered through improved flow predictability and enhanced optimization.

For a technical description of the OpenROAD flow, please refer to our DAC-2019 paper: Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project. The paper is also available from ACM Digital Library. Other publications and presentations are linked here.

CHAPTER

ONE

DOCUMENTATION

The OpenROAD Project has two releases:

- Application (github) (*docs*): The application is a standalone binary for digital place and route that can be used by any other RTL-GDSII flow controller.
- Flow (github) (docs): This is the native OpenROAD flow that consists of a set of integrated scripts for an autonomous RTL-GDSII flow using OpenROAD and other open-source tools.

CHAPTER

TWO

CODE OF CONDUCT

Please read our code of conduct *here*.

CHAPTER

THREE

HOW TO CONTRIBUTE

If you are willing to **contribute**, see the *Getting Involved* section.

If you are a **developer** with EDA background, learn more about how you can use OpenROAD as the infrastructure for your tools in the *Developer Guide* section.

OpenROAD uses Git for version control and contributions. Get familiarised with a quickstart tutorial to contribution *here*.

CHAPTER

FOUR

HOW TO GET IN TOUCH

We maintain the following channels for communication:

- Project homepage and news: https://theopenroadproject.org
- Twitter: https://twitter.com/OpenROAD_EDA
- Issues and bugs:
  - OpenROAD: https://github.com/The-OpenROAD-Project/OpenROAD/issues
- Discussions:
  - OpenROAD: https://github.com/The-OpenROAD-Project/OpenROAD/discussions
- Inquiries: openroad@eng.ucsd.edu

See also our *FAQs*.

CHAPTER

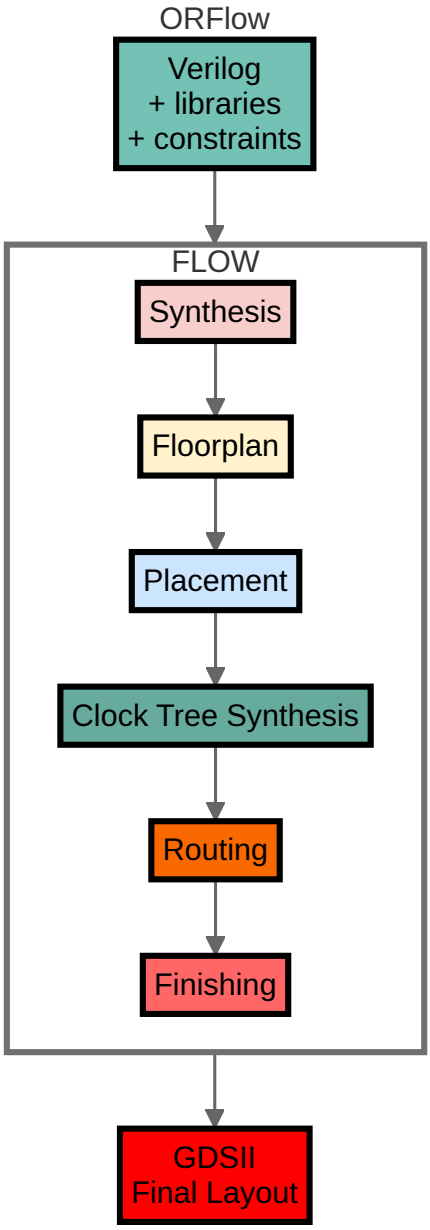
FIVE

SITE MAP

5.1 OpenROAD

5.1.1 About OpenROAD

OpenROAD is the leading open-source, foundational application for semiconductor digital design. The OpenROAD flow delivers an Autonomous, No-Human-In-Loop (NHIL) flow, 24 hour turnaround from RTL-GDSII for rapid design exploration and physical design implementation.


ORFlow
Verilog
+ libraries
+ constraints
FLOW
Synthesis
Floorplan
Placement
Clock Tree Synthesis
Routing
Finishing
GDSII
Final Layout

5.1.2 OpenROAD Mission

OpenROAD eliminates the barriers of cost, schedule risk and uncertainty in hardware design to promote open access to rapid, low-cost IC design software and expertise and system innovation. The OpenROAD application enables flexible flow control through an API with bindings in Tcl and Python.

OpenROAD is used in research and commercial applications such as,

- OpenROAD-flow-scripts from OpenROAD
- OpenLane from Efabless
- Silicon Compiler from Zero ASIC
- Hammer from UC Berkeley
- OpenFASoC from IDEA-FASoC for mixed-signal design flows

OpenROAD fosters a vibrant ecosystem of users through active collaboration and partnership through software development and key alliances. Our growing user community includes hardware designers, software engineers, industry collaborators, VLSI enthusiasts, students and researchers.

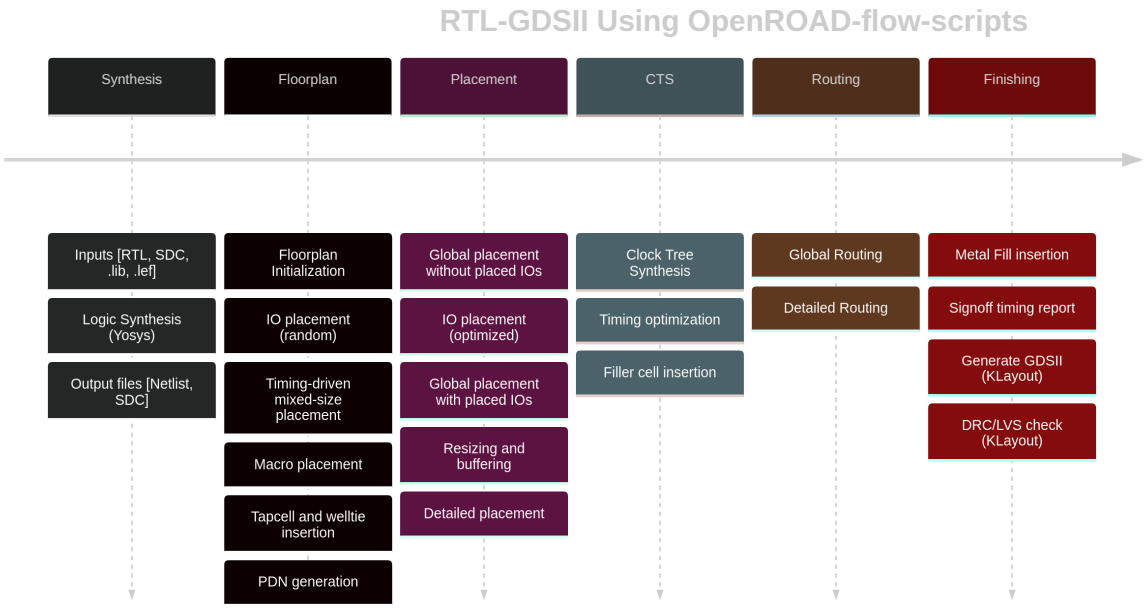
OpenROAD strongly advocates and enables IC design-based education and workforce development initiatives through training content and courses across several global universities, the Google-SkyWater shuttles also includes GlobalFoundries shuttles, design contests and IC design workshops. The OpenROAD flow has been successfully used to date in over 600 silicon-ready tapeouts for technologies up to 12nm.

5.1.3 Getting Started with OpenROAD-flow-scripts

OpenROAD provides OpenROAD-flow-scripts as a native, ready-to-use prototyping and tapeout flow. However, it also enables the creation of any custom flow controllers based on the underlying tools, database and analysis engines. Please refer to the flow documentation here.

OpenROAD-flow-scripts (ORFS) is a fully autonomous, RTL-GDSII flow for rapid architecture and design space exploration, early prediction of QoR and detailed physical design implementation. However, ORFS also enables manual intervention for finer user control of individual flow stages through Tcl commands and Python APIs.

Figure below shows the main stages of the OpenROAD-flow-scripts:



Here are the main steps for a physical design implementation using OpenROAD;

- `Floorplanning`

  - Floorplan initialization - define the chip area, utilization
  - IO pin placement (for designs without pads)
  - Tap cell and well tie insertion
  - PDN- power distribution network creation
- `Global Placement` - Minimize wirelengths
  - Macro placement (RAMs, embedded macros)
  - Standard cell placement
  - Automatic placement optimization and repair for max slew, max capacitance, and max fanout violations and long wires
- `Detailed Placement`
  - Legalize placement - align to grid, adhere to design rules
  - Incremental timing analysis for early estimates
- `Clock Tree Synthesis` - Generate a balanced tree to meet timing and reduce skews
  - Insert buffers and resize for high fanout nets
- `Optimize setup/hold timing`
- `Global routing`
  - Antenna repair
  - Create routing guides
- `Detailed routing`
  - Legalize routes, DRC-correct routing to meet timing, power constraints
- `Chip Finishing`
  - Parasitic extraction using OpenRCX
  - Final timing verification
  - Final physical verification
  - Dummy metal fill for manufacturability
  - Use KLayout or Magic using generated GDS for DRC signoff

GUI

The OpenROAD GUI is a powerful visualization, analysis, and debugging tool with a customizable Tcl interface. The below figures show GUI views for various flow stages including floorplanning, placement congestion, CTS and post-routed design.

Floorplan

Automatic Hierarchical Macro Placement

Placement Congestion Visualization

CTS

Routing

PDK Support

The OpenROAD application is PDK independent. However, it has been tested and validated with specific PDKs in the context of various flow controllers.

OpenLane supports SkyWater 130nm and GlobalFoundries 180nm.

OpenROAD-flow-scripts supports several public and private PDKs including:

Open-Source PDKs

- `GF180` - 180nm
- `SKY130` - 130nm
- `Nangate45` - 45nm
- `ASAP7` - Predictive FinFET 7nm

Proprietary PDKs

These PDKS are supported in OpenROAD-flow-scripts only. They are used to test and calibrate OpenROAD against commercial platforms and ensure good QoR. The PDKs and platform-specific files for these kits cannot be provided due to NDA restrictions. However, if you are able to access these platforms independently, you can create the necessary platform-specific files yourself.

- `GF55` - 55nm
- `GF12` - 12nm
- `Intel22` - 22nm

- `Intel16` - 16nm
- `TSMC65` - 65nm

5.1.4 Tapeouts

OpenROAD has been used for full physical implementation in over 600 tapeouts in SKY130 and GF180 through the Google-sponsored, Efabless MPW shuttle and ChipIgnite programs.

OpenTitan SoC on GF12LP - Physical design and optimization using OpenROAD

Continuous Tapeout Integration into CI

The OpenROAD project actively adds successfully taped out MPW shuttle designs to the CI regression testing. Examples of designs include Open processor cores, RISC-V based SoCs, cryptocurrency miners, robotic app processors, amateur satellite radio transceivers, OpenPower-based Microwatt etc.

5.1.5 Build OpenROAD

To build OpenROAD tools locally in your machine, follow steps from *here*.

5.1.6 Regression Tests

There are a set of executable regression test scripts in `./test/`.

```
# run tests for all tools
./test/regression

# run all flow tests
./test/regression flow

# run <tool> tests
./test/regression <tool>

# run all <tool>-specific unit tests
cd src/<tool>
./test/regression

# run only <TEST_NAME> for <tool>
cd src/<tool>
./test/regression <TEST_NAME>
```

The flow tests check results such as worst slack against reference values. Use `report_flow_metrics [test]...` to see all of the metrics.

```
% report_flow_metrics gcd_nangate45
                       insts    area util slack_min slack_max  tns_max clk_skew max_slew␣
↪max_cap max_fanout DPL ANT drv
gcd_nangate45            368     564  8.8     0.112    -0.015     -0.1    0.004        0␣
↪       0          0   0   0   0
```

To update a failing regression, follow the instructions below:

```
# update log files (i.e. *ok)
save_ok <TEST_NAME>

# update "*.metrics" for tests that use flow test
save_flow_metrics <TEST_NAME>

# update "*.metrics_limits" files
save_flow_metrics_limits <TEST_NAME>
```

5.1.7 Run

```
openroad [-help] [-version] [-no_init] [-exit] [-gui]
         [-threads count|max] [-log file_name] cmd_file
  -help               show help and exit
  -version            show version and exit
  -no_init            do not read .openroad init file
  -threads count|max  use count threads
  -no_splash          do not show the license splash at startup
  -exit               exit after reading cmd_file
  -gui                start in gui mode
  -python             start with python interpreter [limited to db operations]
  -log <file_name>    write a log in <file_name>
  cmd_file            source cmd_file
```

OpenROAD sources the Tcl command file `~/.openroad` unless the command line option `-no_init` is specified.

OpenROAD then sources the command file `cmd_file` if it is specified on the command line. Unless the `-exit` command line flag is specified, it enters an interactive Tcl command interpreter.

A list of the available tools/modules included in the OpenROAD app and their descriptions are available *here*.

5.1.8 Git Quickstart

OpenROAD uses Git for version control and contributions. Get familiarised with a quickstart tutorial to contribution *here*.

5.1.9 Understanding Warning and Error Messages

Seeing OpenROAD warnings or errors you do not understand? We have compiled a table of all messages and you may potentially find your answer here.

5.1.10 License

BSD 3-Clause License. See LICENSE file.

5.1.11 Build OpenROAD

Build

The first step, independent of the build method, is to download the repository:

```
git clone --recursive https://github.com/The-OpenROAD-Project/OpenROAD.git
cd OpenROAD
```

OpenROAD git submodules (cloned by the `--recursive` flag) are located in `src/`.

The default build type is RELEASE to compile optimized code. The resulting executable is in `build/src/openroad`.

Optional CMake variables passed as `-D<var>=<value>` arguments to CMake are show below.

Argument	Value
`CMAKE_BUILD_TYPE`	DEBUG, RELEASE
`CMAKE_CXX_FLAGS`	Additional compiler flags
`TCL_LIBRARY`	Path to Tcl library
`TCL_HEADER`	Path to `tcl.h`
`ZLIB_ROOT`	Path to `zlib`
`CMAKE_INSTALL_PREFIX`	Path to install binary

> **Note:** There is a `openroad_build.log` file that is generated with every build in the build directory. In case of filing issues, it can be uploaded in the "Relevant log output" section of OpenROAD issue forms.

Install dependencies

You may follow our helper script to install dependencies as follows:

```
sudo ./etc/DependencyInstaller.sh
```

> **WARNING**
>
> `etc/DependencyInstaller.sh` defaults to installing system packages and requires sudo access. These packages can affect your environment. We recommend users install dependencies locally using setup.sh from OpenROAD-flow-scripts.

Build Manually

```
mkdir build && cd build
cmake ..
make
make install
```

The default install directory is `/usr/local`. To install in a different directory with CMake use:

```
cmake .. -DCMAKE_INSTALL_PREFIX=<prefix_path>
```

Alternatively, you can use the `DESTDIR` variable with make.

```
make DESTDIR=<prefix_path> install
```

Build using support script

```
./etc/Build.sh
# To build with debug option enabled and if the Tcl library is not on the default path
./etc/Build.sh -cmake="-DCMAKE_BUILD_TYPE=DEBUG -DTCL_LIB=/path/to/tcl/lib"
```

The default install directory is `/usr/local`. To install in a different directory use:

```
./etc/Build.sh -cmake="-DCMAKE_INSTALL_PREFIX=<prefix_path>"
```

LTO Options

By default, OpenROAD is built with link time optimizations enabled. This adds about 1 minute to compile times and improves the runtime by about 11%. If you would like to disable LTO pass `-DLINK_TIME_OPTIMIZATION=OFF` when generating a build.

Build with Address Sanitizer

To enable building with Address Sanitizer, use the argument `-DASAN=ON`. Setting the `ASAN` variable to `ON` adds necessary compile and link options for using Address Sanitizer.

> **Note:** Address Sanitizer adds instrumentation for detecting memory errors. Enabling this option will cause OpenROAD to run slower and consume more RAM.

Build with Prebuilt Binaries

Courtesy of Precision Innovations, there are pre-built binaries of OpenROAD with self-contained dependencies released on a regular basis. Refer to this link here.

5.1.12 Tutorials

OpenROAD Flow Scripts Tutorial

Flow tutorial can be accessed from OpenROAD Flow Scripts documentation here.

5.1.13 Git Quickstart

This tutorial serves as a quickstart to Git and contributing to our repository. If you have not already set up OpenROAD, please follow the instructions *here*.

Tip: This basic tutorial gives instruction for basic password Git authentication. If you would like to setup SSH authentication, please follow this guide.

Forking

You will need your own fork to work on the code. Go to the `OpenROAD` project page and hit the `Fork` button. You will want to clone your fork to your machine:

```
git clone https://github.com/your-user-name/OpenROAD.git
cd OpenROAD
git remote add upstream https://github.com/The-OpenROAD-Project/OpenROAD.git
git fetch upstream
```

This creates the directory `OpenROAD` and connects your repository to the upstream (master project) *OpenROAD* repository.

Creating a branch

You want your master branch to reflect only production-ready code, so create a feature branch for making your changes. For example:

```
git checkout master && git branch shiny-new-feature
git checkout shiny-new-feature
# Or equivalently,
git checkout master && checkout -b shiny-new-feature
```

This changes your working directory to the shiny-new-feature branch. Keep any changes in this branch specific to one bug or feature so it is clear what the branch brings to OpenROAD. You can have many shiny-new-features and switch in between them using the git checkout command.

When creating this branch, make sure your master branch is up to date with the latest upstream master version. To update your local master branch, you can do:

```
git checkout master
git pull upstream master
```

When you want to update the feature branch with changes in master after you created the branch, check the section on *updating a PR*.

Committing your code

Keep style fixes to a separate commit to make your pull request more readable. Once you've made changes, you can see them by typing:

```
git status
```

If you have created a new file, it is not being tracked by git. Add it by typing:

```
git add path/to/file-to-be-added.py
```

Doing `git status` again should give something like:

```
# On branch shiny-new-feature
#
#       modified:   /relative/path/to/file-you-added.py
#
```

Finally, commit your changes to your local repository with an explanatory commit message. Do note the `-s` option is needed for developer signoff.

```
git commit -s -m "your commit message goes here"
```

Pushing your changes

When you want your changes to appear publicly on your GitHub page, push your forked feature branch's commits:

```
git push origin shiny-new-feature
```

Here `origin` is the default name given to your remote repository on GitHub. You can see the remote repositories:

```
git remote -v
```

If you added the upstream repository as described above you will see something like:

```
origin  https://github.com/your-user-name/OpenROAD.git (fetch)
origin  https://github.com/your-user-name/OpenROAD.git (push)
upstream        https://github.com/The-OpenROAD-Project/OpenROAD.git (fetch)
upstream        https://github.com/The-OpenROAD-Project/OpenROAD.git (push)
```

Now your code is on GitHub, but it is not yet a part of the OpenROAD project. For that to happen, a pull request needs to be submitted on GitHub.

Review your code

When you're ready to ask for a code review, file a pull request. Before you do, once again make sure that you have followed all the guidelines outlined in the *Developer's Guide* regarding code style, tests, performance tests, and documentation. You should also double check your branch changes against the branch it was based on:

1. Navigate to your repository on GitHub – https://github.com/your-user-name/OpenROAD
2. Click on `Branches`
3. Click on the `Compare` button for your feature branch

4. Select the `base` and `compare` branches, if necessary. This will be `master` and `shiny-new-feature`, respectively.

Submitting the pull request

If everything looks good, you are ready to make a pull request. A pull request is how code from a local repository becomes available to the GitHub community and can be looked at and eventually merged into the master version. This pull request and its associated changes will eventually be committed to the master branch and available in the next release. To submit a pull request:

1. Navigate to your repository on GitHub
2. Click on the `Compare & pull request` button
3. You can then click on `Commits` and `Files Changed` to make sure everything looks okay one last time
4. Write a description of your changes in the `Preview Discussion` tab
5. Click `Send Pull Request`.

This request then goes to the repository maintainers, and they will review the code.

Updating your pull request

Based on the review you get on your pull request, you will probably need to make some changes to the code. In that case, you can make them in your branch, add a new commit to that branch, push it to GitHub, and the pull request will be automatically updated. Pushing them to GitHub again is done by:

```
git push origin shiny-new-feature
```

This will automatically update your pull request with the latest code and restart the *Continuous Integration* tests.

Another reason you might need to update your pull request is to solve conflicts with changes that have been merged into the master branch since you opened your pull request.

To do this, you need to `merge upstream master` in your branch:

```
git checkout shiny-new-feature
git fetch upstream
git merge upstream/master
```

If there are no conflicts (or they could be fixed automatically), a file with a default commit message will open, and you can simply save and quit this file.

If there are merge conflicts, you need to solve those conflicts. See this article for an explanation on how to do this. Once the conflicts are merged and the files where the conflicts were solved are added, you can run `git commit` to save those fixes.

If you have uncommitted changes at the moment you want to update the branch with master, you will need to `stash` them prior to updating.

See also:

See the stash docs.

This will effectively store your changes and they can be reapplied after updating.

After the feature branch has been updated locally, you can now update your pull request by pushing to the branch on GitHub:

```
git push origin shiny-new-feature
```

Tips for a successful pull request

If you have made it to the `Review your code` phase, one of the core contributors may take a look. Please note however that a handful of people are responsible for reviewing all of the contributions, which can often lead to bottlenecks.

To improve the chances of your pull request being reviewed, you should:

- **Reference an open issue** for non-trivial changes to clarify the PR's purpose
- **Ensure you have appropriate tests**. These should be the first part of any PR
- **Keep your pull requests as simple as possible**. Larger PRs take longer to review
- **Ensure that CI is in a green state**. Reviewers may not even look otherwise
- **Keep updating your pull request**, either by request or every few days

Acknowledgements

This page has been adapted from pandas Developer Guide.

5.2 OpenROAD

OpenROAD is run using Tcl scripts. The following commands are used to read and write design data.

```
read_lef [-tech] [-library] filename
read_def filename
write_def [-version 5.8|5.7|5.6|5.5|5.4|5.3] filename
read_verilog filename
write_verilog filename
read_db filename
write_db filename
write_abstract_lef filename
```

Use the Tcl `source` command to read commands from a file.

```
source [-echo] file
```

If an error is encountered in a command while reading the command file, then the error is printed and no more commands are read from the file. If `file_continue_on_error` is 1 then OpenROAD will continue reading commands after the error.

If `exit_on_error` is 1 then OpenROAD will exit when it encounters an error.

OpenROAD can be used to make a OpenDB database from LEF/DEF, or Verilog (flat or hierarchical). Once the database is made it can be saved as a file with the `write_db` command. OpenROAD can then read the database with the `read_db` command without reading LEF/DEF or Verilog.

The `read_lef` and `read_def` commands can be used to build an OpenDB database as shown below. The `read_lef -tech` flag reads the technology portion of a LEF file. The `read_lef -library` flag reads the MACROs in the LEF file. If neither of the `-tech` and `-library` flags are specified they default to `-tech -library` if no technology has been read and `-library` if a technology exists in the database.

```
read_lef liberty1.lef
read_def reg1.def
# Write the db for future runs.
write_db reg1.db
```

The `read_verilog` command is used to build an OpenDB database as shown below. Multiple Verilog files for a hierarchical design can be read. The `link_design` command is used to flatten the design and make a database.

```
read_lef liberty1.lef
read_verilog reg1.v
link_design top
# Write the db for future runs.
write_db reg1.db
```

5.2.1 Example scripts

Example scripts demonstrating how to run OpenROAD on sample designs can be found in /test. Flow tests taking sample designs from synthesizable RTL Verilog to detail-routed final layout in the open-source technologies Nangate45 and Sky130HD are shown below.

```
gcd_nangate45.tcl
aes_nangate45.tcl
tinyRocket_nangate45.tcl
gcd_sky130hd.tcl
aes_sky130hd.tcl
ibex_sky130hd.tcl
```

Each of these designs use the common script `flow.tcl`.

5.2.2 Abstract LEF Support

OpenROAD contains an abstract LEF writer that can take your current design and emit an abstract LEF representing the external pins of your design and metal obstructions.

```
write_abstract_lef (-bloat_factor bloat_factor|-bloat_occupied_layers) \
                   filename
```

Options

Switch Name	Description
`-bloat_factor`	Specifies the bloat factor used when bloating then merging shapes into LEF obstructions. The factor is measured in # of default metal pitches for the respective layer. A factor of `0` will result in detailed LEF obstructions
`-bloat_occupied_layers`	Generate over obstructions (obstructions over the entire layer) for each layer where shapes are present

Examples

```
read reg1.db

# Bloat metal shapes by 3 pitches (respectively for every layer) and then merge
write_abstract_lef -bloat_factor 3 reg1_abstract.lef

# Produce cover obstructions for each layer with shapes present
write_abstract_lef -bloat_occupied_layers reg1_abstract.lef
```

Global Connections

Add global connections

The `add_global_connection` command is used to specify how to connect power and ground pins on design instances to the appropriate supplies.

```
add_global_connection -net net_name \
                      [-inst_pattern inst_regular_expression] \
                      -pin_pattern pin_regular_expression \
                      (-power|-ground) \
                      [-region region_name]
```

Options

Switch Name	Description
`-net`	Specifies the name of the net in the design to which connections are to be added
`-inst_pattern`	Optional specifies a regular expression to select a set of instances from the design. (Default: .*)
`-pin_pattern`	Species a regular expression to select pins on the selected instances to connect to the specified net
`-power`	Specifies that the net it a power net
`-ground`	Specifies that the net is a ground net
`-region`	Specifies the name of the region for this rule

Examples

```
# Stdcell power/ground pins
add_global_connection -net VDD -pin_pattern {^VDD$} -power
add_global_connection -net VSS -pin_pattern {^VSS$} -ground

# SRAM power ground pins
add_global_connection -net VDD -pin_pattern {^VDDPE$}
add_global_connection -net VDD -pin_pattern {^VDDCE$}
add_global_connection -net VSS -pin_pattern {^VSSE$}
```

Perform global connections

The `global_connect` command is used to connect power and ground pins on design instances to the appropriate supplies.

```
global_connect
```

Clear global connection rules

The `clear_global_connect` command is used remove all defined global connection rules.

```
clear_global_connect
```

Report global connection rules

The `report_global_connect` command is used print out the currently defined global connection rules.

```
report_global_connect
```

5.2.3 TCL functions

Get the die and core areas as a list in microns: `llx lly urx ury`

```
ord::get_die_area
ord::get_core_area
```

5.2.4 FAQs

Check out GitHub discussion about this tool.

5.2.5 License

BSD 3-Clause License.

5.2.6 OpenDB

OpenDB is a design database to support tools for physical chip design. It was originally developed by Athena Design Systems. Nefelus, Inc. acquired the rights to the code and open-sourced it with BSD-3 license in 2019 to support the DARPA OpenROAD project.

The structure of OpenDB is based on the text file formats LEF (library) and DEF (design) formats version 5.6. OpenDB supports a binary file format to save and load the design much faster than using LEF and DEF.

OpenDB is written in C++ 98 with standard library style iterators. The classes are designed to be fast enough to base an application on without having to copy them into application-specific structures.

Directory structure

```
include/odb/db.h - public header for all database classes
src/db - private/internal database representations
src/lefin - LEF reader
src/lefout - LEF writer
src/defin - DEF reader
src/defout - DEF writer
```

Database API

We are still working on documenting the APIs. We have over 1,800 objects and functions that we are still documenting (for both TCL and Python). **Contributions are very welcome in this effort**. Find starting points below.

TCL

After building successfully, run OpenDB Tcl shell using `../../build/src/odb/src/swig/tcl/odbtcl`. An example usage:

```
set db [dbDatabase_create]
set lef_parser [new_lefin $db true]
set tech [lefin_createTech $lef_parser ./src/odb/test/data/gscl45nm.lef]
```

You can find examples on using the API from Tcl under `test/tcl/` directory.

The full set of the Tcl commands exposed can be found under `./build/src/swig/tcl/opendb_wrapper.cpp`. Search for `SWIG_prefix`.

Python

After building successfully, run `openroad -python` to enable the Python interpreter. You can find examples on using the API from Python under `test/python/` directory.

To list the full set of the Python classes exposed run `openroad -python` then:

```
import openroad
import odb
print(', '.join(dir(openroad)))
print(', '.join(dir(odb)))
```

C++

All public database classes are defined in `db.h`. These class definitions provide all functions for examining and modifying the database objects. The database is an object itself, so multiple database objects can exist simultaneously (no global state).

`dbTypes.h` defines types returned by database class member functions.

All database objects are in the `odb` namespace.

- `dbChip`

- dbBlock
- dbTech
- dbLib

All database objects have a 32bit object identifier accessed with the dbObject::getOID base class member function that returns a uint. This identifier is preserved across save/restores of the database so it should be used to reference database object by data structures instead of pointers if the reference lifetime is across database save/restores. OIDs allow the database to have exactly the same layout across save/restores.

The database distance units are **nanometers** and use the type uint.

Example scripts

Regression tests

There are a set of regression tests in /test.

```
./test/regression-tcl.sh
./test/regression-py.sh
```

Database Internals

The internal description included here is paraphrased from Lukas van Ginneken by James Cherry.

The database separates the implementation from the interface, and as a result, each class becomes two classes, a public one and a private one. For instance, dbInst has the public API functions, while class _dbInst has the private data fields.

The objects are allocated in dynamically resizable tables, the implementation of which is in dbTable.hpp. Each table consists of a number of pages, each containing 128 objects. The table contains the body of the struct, not a set of pointers. This eliminates most of the pointer overhead while iteration is accomplished by stepping through the table. Thus, grouping these objects does not require a doubly-linked list and saves 16 bytes per object (at the cost of some table overhead). Each object has an id, which is the index into the table. The lowest 7 bits are the index in the page, while the higher bits are the page number. Object id's are persistent when saving and reading the data model to disk, even as pointer addresses may change.

Everything in the data model can be stored on disk and restored from disk exactly the way it was. An extensive set of equality tests and diff functions make it possible to check for even the smallest deviation. The capability to save an exact copy of the state of the system makes it possible to create a checkpoint. This is a necessary capability for debugging complex systems.

The code follows the definition of LEF and DEF closely and reflects many of the idiosyncrasies of LEF and DEF. The code defines many types of objects to reflect LEF and DEF constructs although it sometimes uses different terminology, for instance, the object to represent a library cell is called dbMaster while the LEF keyword is MACRO.

The data model supports the EEQ and LEQ keywords (i.e., electrically equivalent and logically equivalent Masters), which could be useful for sizing. However, it does not support any logic function representation. In general, there is very limited support for synthesis-specific information: no way to represent busses, no way to represent logic function, very limited understanding of signal flow, limited support of timing information, and no support for high level synthesis or test insertion.

The db represents routing as in DEF, representing a trace from point to point with a given width. The layout for a net is stored in a class named dbWire and it requires a special dbWireDecoder (which works like an iterator) to unpack the data and another dbWireEncoder to pack it. The data model does not support a region query and objects that are in the same layer are scattered about the data model and are of different classes.

This means that whatever tool is using the layout information will have to build its own data structures that are suitable to the layout operations of that tool. For instance, the router, the extractor, and the DRC engine would each have to build their unique data structures. This encourages batch mode operation (route the whole chip, extract the whole chip, run DRC on the whole chip).

Limitations

FAQs

Check out GitHub discussion about this tool.

LICENSE

BSD 3-Clause License. See LICENSE file.

Automatic Code Generator

This is an automatic code generation tool for OpenDB objects and Iterators. To test the tool you can use the following command

```
python3 gen.py --json schema.json --src_dir ../db --include_dir ../../include/odb --
↪templates templates
```

Where schema.json is the json file that includes the requirements, src is the source files directory, include is the include directory, and templates is the directory including the jinja templates for the classes.

Empty sections are removed by default from the output. If you need to add someting to a section that is currently empty, you can run the generator with –keep_empty to preserve them. Once the section is filled in, the flag can be dropped and the code regnerated to remove the remaining empty sections.

Python Unit Tests

Running tests

For running the python unit tests you will need to install first *testtools* and *unittest-parallel* which enables concurrent unit testing

```
pip3 install testtools
pip3 install unittest-parallel
```

Then, you can run the unit tests in sequence by running

```
../unitTests.sh
```

or in parallel by running

```
../unitTests.sh parallel
```

Note: The test cases within each Unit Test run in parallel in both situations

Tests Structure

The directory unitTestsPython includes unit tests for OpenDB Python APIs. Any test file starts with 'Test' followed by the test target.

odbUnitTest.py:

This includes `TestCase` class which inherits from `unittest.TestCase` with additional functionalities:

- `changeAndTest(self,obj,SetterName,GetterName,expectedVal,*args)`which is a function for changing a value and testing for the effect of that change where:
  - `obj` is the object to be tested
  - `SetterName` is the name of the function to be called for changing a value
  - `GetterName` is the name of the function to be called for testing the effect
  - `expectedVal` is the expected value for the testing
  - `*args` are the arguments passed to the `SetterName` function

  So, in the end, the expected behavior is:

```
obj.SetterName(*args)

assert(obj.GetterName()==expectedVal)
```

- `check(self,obj,GetterName,expectedVal,*args)` which tests against expected value
- `change(self,obj,SetterName,*args)` which changes a value in the object
- `main()` runs the `TestCase` in sequential order
- `mainParallel(Test)` runs the passed `Test` class in parallel

helper.py:

A set of functions for creating simple db instances to be used for testing. You can find the description of each function in the comments

TestNet.py:

Unit test class for testing dbNet. It inherits from `odbUnitTest.TestCase` . it consists of

- `setUp(self)` function to be called before each test case. Here, we create the database with the desired chip, block, masters, instances and nets.
- `tearDown(self)` function to be called after each test case. Here, we destroy our db.
- `test_*(self)` the test cases functions. Their names should start with `test` for the unittest suite to recognize.

TestDestroy.py:

Integration test class for testing the `destroy(*args)` function on OpenDB.

- `test_destroy_net` destroying net and testing for the effect on the *block,inst, iterms and bterms*
- `test_destroy_inst` destroying instance and testing for the effect on *block, iterms, net, bterms*
- `test_destroy_bterm` destroying bterm and testing for the effect on *block and net*
- `test_destroy_block` destroying block and testing for the effect on *block(parent and child relation), and chip*
- `test_destroy_bpin` destroying bpin and testing for the effect on *bterm*
- `test_create_destroy_wire` destroying wire and test for the effect on *net*
- `test_destroy_capnode` destroying capnode and test for the effect on *net(node and connected ccsegs)*
- `test_destroy_ccseg` destroying ccseg and test for the effect on *node,block and net*
- `test_destroy_lib` destroying lib and test for the effect on *db*
- `test_destroy_obstruction` destroying obstruction and test for the effect on *block*
- `test_create_regions` creating regions and test for the effect on *block and region(parent and child relation)*
- `test_destroy_region_child` destroying _ and test for the effect on *block and region(parent)*
- `test_destroy_region_parent` destroying _ and test for the effect on *block*

TestBlock.py:

Unit Test for dbBlock

- `test_find` testing the find function with *BTerm, Child, Inst, Net, ITerm, ExtCornerBlock, nonDefaultRule, Region*
- Testing the ComputeBBox() function through the first call of getBBox:
  - `test_bbox0` testing empty block box
  - `test_bbox1` testing block box with Inst placed
  - `test_bbox2` testing block box with Inst and BPin placed
  - `test_bbox3` testing block box with Inst, BPin and Obstruction placed
  - `test_bbox3` testing block box with Inst, BPin, Obstruction and SWire placed

TestBTerm.py:

Unit Test for dbBTerm

- `test_idle` testing for idle disconnected `BTerm` behavior
- `test_connect` testing connect function of `BTerm` on `BTerm` and `Net`
- `test_disconnect` testing disconnect function of `BTerm` on `BTerm` and `Net`

TestInst.py:

Unit Test for dbInst

- `test_swap_master` testing swap master function

TestITerm.py:

Unit Test for dbITerm

- `test_idle` testing for disconnected ITerm without a net
- `test_connection_from_iterm` testing the connect(ITerm,...) and disconnect functions of ITerm and their effect on ITerm and Net
- `test_connection_from_inst` testing the connect(Inst,...) and disconnect functions of ITerm and their effect on ITerm and Net
- Testing for getAvgXY() function
  - `test_avgxy_R0` testing with default orientation R0
  - `test_avgxy_R90` testing with different orientation R90 for transformation

---

Problems Found In Testing

- multiple core dumps that leads to aborting the process:
  - dbNet.get1st*() (when nothing on top of the list)
  - childRegion.getParent() (after destroying the parent region)
- Implementation of ComputeBBox() is flawed and needs to be reconsidered

Adding new fields in DB Object

For example `add_pitchDiag` in object `DbTechLayer`.

	Action	File	Source Code
1	Add Fields at the .h file	dbTechLayer.h	
2	Define a keyword for db rev number	dbDatabase.h	#define ADS_DB_DF58 52
3	Set the current rev number same as	dbDatabase.h	#define ADS_DB_SCHEMA_MINOR 52
4	Stream in new fields Conditionally upon Schema number	dbTechLayercpp	if (stream.getDatabase()->isSchema(ADS_DB_DF58)) { stream >> layer._pitchDiag;
5	Stream out new fields Conditionally upon Schema number	dbTechLayercpp	if (stream.getDatabase()->isSchema(ADS_DB_DF58)) { stream << layer._pitchDiag;
6	Conditionally Diff new fields	dbTechLayercpp	if (stream.getDatabase()->isSchema(ADS_DB_DF58)) { DIFF_FIELD(_pitchDiag);
7	Conditionally Diff Out new fields	dbTechLayercpp	if (stream.getDatabase()->isSchema(ADS_DB_DF58)) { DIFF_OUT_FIELD(_pitchDiag);
8	Created access APIs to the fields	dbTechLayercpp	"dbTechLayer::getPitchDiag(), dbTechLayer::setPitchDiag(int pitch)"
9	Add new APIs in include/db.h	db.h	class dbTechLayer

5.2.7 Graphical User Interface

The graphical user interface can be access by launching OpenROAD with `-gui` or by opening it from the command-line with `gui::show`.

Commands

Add buttons to the toolbar

```
create_toolbar_button [-name name]
                      -text button_text
                      -script tcl_script
                      [-echo]
```

Returns: name of the new button, either `name` or `buttonX`.

Options description:

- `button_text`: The text to put on the button.
- `tcl_script`: The tcl script to evaluate when the button is pressed.
- `name`: (optional) name of the button, used when deleting the button.
- `echo`: (optional) indicate that the commands in the `tcl_script` should be echoed in the log.

To remove the button:

```
gui::remove_toolbar_button name
```

Add items to the menubar

```
create_menu_item [-name name]
                 [-path menu_path]
                 -text item_text
                 -script tcl_script
                 [-shortcut key_shortcut]
                 [-echo]
```

Returns: name of the new item, either `name` or `actionX`.

Options description:

- `item_text`: The text to put on the item.
- `tcl_script`: The tcl script to evaluate when the button is pressed.
- `name`: (optional) name of the item, used when deleting the item.
- `menu_path`: (optional) Menu path to place the new item in (hierarchy is separated by /), defaults to "Custom Scripts", but this can also be "Tools" or "New menu/New submenu".
- `key_shortcut`: (optional) key shortcut to trigger this item.
- `echo`: (optional) indicate that the commands in the `tcl_script` should be echoed in the log.

To remove the item:

```
gui::remove_menu_item name
```

Save screenshot of layout

This command can be both be used when the GUI is active and not active.

```
save_image [-resolution microns_per_pixel]
           [-area {x0 y0 x1 y1}]
           [-width width]
           [-display_option {option value}]
           filename
```

Options description:

- `filename` path to save the image to.
- `x0, y0` first corner of the layout area (in microns) to be saved, default is to save what is visible on the screen unless called when gui is not active and then it selected the whole block.
- `x1, y1` second corner of the layout area (in microns) to be saved, default is to save what is visible on the screen unless called when gui is not active and then it selected the whole block.
- `microns_per_pixel` resolution in microns per pixel to use when saving the image, default will match what the GUI has selected.
- `width` width of the output image in pixels, default will be computed from the resolution. Cannot be used with `-resolution`.
- `option` specific setting for a display option to show or hide specific elements. For example, to hide metal1 `-display_option {Layers/metal1 false}`, to show routing tracks `-display_option {Tracks/Pref true}`, or to show everthing `-display_option {* true}`.

Save screenshot of clock trees

```
save_clocktree_image filename
                     -clock clock_name
                     [-width width]
                     [-height height]
                     [-corner corner]
```

Options description:

- `filename` path to save the image to.
- `-clock` name of the clock to save the clocktree for.
- `-corner` name of the timing corner to save the clocktree for, default to the first corner defined.
- `-height` height of the image in pixels, defaults to the height of the GUI widget.
- `-width` width of the image in pixels, defaults to the width of the GUI widget.

Selecting objects

```
select -type object_type
       [-name glob_pattern]
       [-filter attribute=value]
       [-case_insensitive]
       [-highlight group]
```

Returns: number of objects selected.

Options description:

- `object_type`: name of the object type. For example, `Inst` for instances, `Net` for nets, and `DRC` for DRC violations.
- `glob_pattern`: (optional) filter selection by the specified name. For example, to only select clk nets `*clk*`. Use `-case_insensitive` to filter based on case insensitive instead of case sensitive.
- `attribute=value`: (optional) filter selection based on the objects' properties. `attribute` represents the property's name and `value` the property's value. In case the property holds a collection (e. g. BTerms in a Net) or a table (e. g. Layers in a Generate Via Rule) `value` can be any element within those. A special case exists for checking whether a collection is empty or not by using the value `CONNECTED`. This can be useful to select a specific group of elements (e. g. BTerms=CONNECTED will select only Nets connected to Input/Output Pins).
- `group`: (optional) add the selection to the specific highlighting group. Values can be 0 to 7.

Displaying timing cones

```
display_timing_cone pin
                    [-fanin]
                    [-fanout]
                    [-off]
```

Options description:

- `pin`: name of the instance or block pin.

- `fanin`: (optional) display the fanin timing cone.
- `fanout`: (optional) display the fanout timing cone.
- `off`: (optional) remove the timing cone.

Limit drawing to specific nets

```
focus_net net
          [-remove]
          [-clear]
```

Options description:

- `pin`: name of the net.
- `remove`: (optional) removes the net from from the focus.
- `clear`: (optional) clears all nets from focus.

TCL functions

Support

Determine is the GUI is active:

```
gui::enabled
```

Announce to the GUI that a design was loaded (note: this is only needed when the design was loaded through the odb API and not via `read_def` or `read_db`):

```
gui::design_created
```

To load the results of a DRC report:

```
gui::load_drc filename
```

Opening and closing

To open the GUI from the command-line (this command does not return until the GUI is closed):

```
gui::show
gui::show script
gui::show script interactive
```

Options description:

- `script` TCL script to evaluate in the GUI.
- `interactive` indicates if true the GUI should open in an interactive session (default), or if false that the GUI would execute the script and return to the terminal.

To close the GUI and return to the command-line:

```
gui::hide
```

Layout navigation

To fit the whole layout in the window:

```
gui::fit
```

To zoom in our out to a specific region:

```
gui::zoom_to x0 y0 x1 y1
```

Options description:

- `x0, y0` first corner of the layout area in microns.
- `x1, y1` second corner of the layout area in microns.

To zoom in the layout:

```
gui::zoom_in
gui::zoom_in x y
```

Options description:

- `x, y` new center of layout.

To zoom out the layout:

```
gui::zoom_out
gui::zoom_out x y
```

Options description:

- `x, y` new center of layout.

To move the layout to new area:

```
gui::center_at x y
```

Options description:

- `x, y` new center of layout.

To change the resolution to a specific value:

```
gui::set_resolution resolution
```

Options description:

- `resolution` database units per pixel.

Selections

To add a single net to the selected items:

```
gui::selection_add_net name
```

Options description:

- `name` name of the net to add.

To add several nets to the selected items:

```
gui::selection_add_nets name_regex
```

Options description:

- `name_regex` regular expression of the net names to add.

To add a single instance to the selected items:

```
gui::selection_add_inst name
```

Options description:

- `name` name of the instance to add.

To add several instances to the selected items:

```
gui::selection_add_insts name_regex
```

Options description:

- `name_regex` regular expression of the instance names to add.

To add items at a specific point or in an area:

```
gui::select_at x y
gui::select_at x y append
gui::select_at x0 y0 x1 y1
gui::select_at x0 y0 x1 y1 append
```

Options description:

- `x, y` point in the layout area in microns.
- `x0, y0` first corner of the layout area in microns.
- `x1, y1` second corner of the layout area in microns.
- `append` if `true` (the default value) append the new selections to the current selection list, else replace the selection list with the new selections.

To navigate through multiple selected items:

```
gui::select_next
gui::select_previous
```

Returns: current index of the selected item.

To clear the current set of selected items:

```
gui::clear_selections
```

To get the properties for the current selection in the Inspector:

```
gui::get_selection_property name
```

Options description:

- `name` name of the property. For example, `Type` for object type or `bbox` for the bounding box of the object.

To animate the current selection in the Inspector:

```
gui::selection_animate
gui::selection_animate repeat
```

Options description:

- `repeat`: indicate how many times the animation should repeat, default value is 0 repeats. If the value is 0, the animation will repeat indefinitely.

Highlighting

To highlight a net:

```
gui::highlight_net name
gui::highlight_net name highlight_group
```

Options description:

- `name` name of the net to highlight.
- `highlight_group` group to add the highlighted net to, defaults to `0`, valid groups are `0 - 7`.

To highlight an instance:

```
gui::highlight_inst name
gui::highlight_inst name highlight_group
```

Options description:

- `name` name of the instance to highlight.
- `highlight_group` group to add the highlighted instance to, defaults to `0`, valid groups are `0 - 7`.

To clear the highlight groups:

```
gui::clear_highlights
gui::clear_highlights highlight_group
```

Options description:

- `highlight_group` group to clear, defaults to `0`, valid groups are `-1 - 7`. Use `-1` to clear all groups.

Rulers

To add a ruler to the layout:

1. either press `k` and use the mouse to place it visually. To disable snapping for the ruler when adding, hold the `Ctrl` key, and to allow non-horizontal or vertical snapping when completing the ruler hold the `Shift` key.
2. or use the command:

```
gui::add_ruler x0 y0 x1 y1
gui::add_ruler x0 y0 x1 y1 label
gui::add_ruler x0 y0 x1 y1 label name
gui::add_ruler x0 y0 x1 y1 label name euclidian
```

Returns: name of the newly created ruler.

Options description:

- `x0, y0` first end point of the ruler in microns.
- `x1, y1` second end point of the ruler in microns.
- `label` text label for the ruler.
- `name` name of the ruler.
- `euclidian` 1 for euclidian ruler, and `0` for regular ruler.

To remove a single ruler:

```
gui::delete_ruler name
```

Options description:

- `name` name of the ruler.

To remove all the rulers:

```
gui::clear_rulers
```

Heat Maps

The currently availble heat maps are:

- `Power`
- `Routing`
- `Placement`
- `IRDrop`

To control the settings in the heat maps:

```
gui::set_heatmap name option
gui::set_heatmap name option value
```

Options description:

- `name` is the name of the heatmap.
- `option` is the name of the option to modify. If option is `rebuild` the map will be destroyed and rebuilt.

- `value` is the new value for the specified option. This is not used when rebuilding map.

These options can also be modified in the GUI by double-clicking the underlined display control for the heat map.

To save the raw data from the heat maps ins a comma separated value (CSV) format:

```
gui::dump_heatmap name filename
```

Options description:

- `name` is the name of the heatmap.
- `filename` path to the file to write the data to.

GUI Display Controls

Control the visible and selected elements in the layout:

```
gui::set_display_controls name display_type value
```

Options description:

- `name` is the name of the control. For example, for the power nets option this would be `Signals/Power` or could be `Layers/*` to set the option for all the layers.
- `display_type` is either `visible` or `selectable`
- `value` is either `true` or `false`

To check the visibility or selectability of elements in the layout:

```
gui::check_display_controls name display_type
```

Options description:

- `name` is the name of the control. For example, for the power nets option this would be `Signals/Power` or could be `Layers/*` to set the option for all the layers.
- `display_type` is either `visible` or `selectable`

When performing a batch operation changing the display controls settings, the following commands can be used to save the current state of the display controls and restore them at the end.

```
gui::save_display_controls
gui::restore_display_controls
```

GUI Controls

To request user input via the GUI:

```
gui::input_dialog title question
```

Returns: a string with the input, or empty string if canceled.

Options description:

- `title` is the title of the input message box.
- `question` is the text for the message box.

Pause the execution of the script:

```
gui::pause
gui::pause timeout
```

Options description:

- `timeout` is specified in milliseconds, if it is not provided the pause will last until the user presses the Continue button.

To open or close a specific layout widget:

```
gui::show_widget name
gui::hide_widget name
```

Options description:

- `name` of the widget. For example, the display controls would be "Display Control".

License

BSD 3-Clause License. See LICENSE file.

5.2.8 TritonPart : An Open-Source Constraints-Driven Partitioner

TritonPart is an open-source constraints-driven partitioner. It can be used to partition a hypergraph or a gate-level netlist.

Features

- Start of the art multiple-constraints driven partitioning "multi-tool"
- Optimizes cost function based on user requirement
- Permissive open-source license
- Solves multi-way partitioning with following features:
  - Multidimensional real-value weights on vertices and hyperedges
  - Multilevel coarsening and refinement framework
  - Fixed vertices constraint
  - Timing-driven partitioning framework
  - Group constraint: Groups of vertices need to be in same block
  - Embedding-aware partitioning

Dependency

We use Google OR-Tools as our ILP solver. Please install Google OR-Tools following the instructions.

How to partition a hypergraph in the way you would using hMETIS (min-cut partitioning)

```
triton_part_hypergraph -hypergraph_file des90.hgr -num_parts 5 -balance_constraint 2 -
↪seed 2
```

You can also check the provided example *here*.

How to perform the embedding-aware partitioning

```
set num_parts 2
set balance_constraint 2
set seed 0
set design sparcT1_chip2
set hypergraph_file "${design}.hgr"
set placement_file "${design}.hgr.ubfactor.2.numparts.2.embedding.dat"
set solution_file "${design}.hgr.part.${num_parts}"

triton_part_hypergraph  -hypergraph_file $hypergraph_file -num_parts $num_parts \
                        -balance_constraint $balance_constraint \
                        -seed $seed  \
                        -placement_file ${placement_file} -placement_wt_factors { 0.
↪00005 0.00005 } \
                        -placement_dimension 2
```

You can find the provided example *here*.

How to partition a netlist

```
# set technology information
set ALL_LEFS “list_of_lefs”
set ALL_LIBS “list_of_libs”
# set design information
set design “design_name”
set top_design “top_design”
set netlist “netlist.v”
set sdc “timing.sdc”
foreach lef_file ${ALL_LEFS} {
  read_lef $lef_file
}
foreach lib_file ${ALL_LIBS} {
  read_lib $lib_file
}
read_verilog $netlist
link_design $top_design
read_sdc $sdc
```

(continues on next page)

(continued from previous page)

```
set num_parts 5
set balance_constraint 2
set seed 0
set top_n 100000
# set the extra_delay_cut to 20% of the clock period
# the extra_delay_cut is introduced for each cut hyperedge
set extra_delay_cut 9.2e-10
set timing_aware_flag true
set timing_guardband true
set part_design_solution_file "${design}_part_design.hgr.part.${num_parts}"

################################################################################################
↪#####
### TritonPart with slack progagation
################################################################################################
↪#####
puts "Start TritonPart with slack propagation"
# call triton_part to partition the netlist
triton_part_design -num_parts $num_parts -balance_constraint $balance_constraint \
                   -seed $seed -top_n $top_n \
                   -timing_aware_flag $timing_aware_flag -extra_delay $extra_delay_cut \
                   -guardband_flag $timing_guardband \
                   -solution_file $part_design_solution_file
```

You can find the provided example *here*.

License

BSD 3-Clause License. See LICENSE file.

5.2.9 Restructure

The restructure module in OpenROAD (`rmp`) is based on an interface to ABC for local resynthesis. The package allows logic restructuring that targets area or timing. It extracts a cloud of logic using the OpenSTA timing engine, and passes it to ABC through `blif` interface. Multiple recipes for area or timing are run to obtain multiple structures from ABC; the most desirable among these is used to improve the netlist. The ABC output is read back by a `blif` reader which is integrated to OpenDB. `blif` writer and reader also support constants from and to OpenDB. Reading back of constants requires insertion of tie cells which should be provided by the user as per the interface described below.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Restructuring can be done in two modes: area or delay.

Area Mode

```
restructure
    -liberty_file liberty_file
    -target area
    -tielo_pin  tielo_pin_name
    -tiehi_pin  tiehi_pin_name
```

Timing Mode

```
restructure
    -liberty_file liberty_file
    -target delay
    -slack_threshold slack_val
    -depth_threshold depth_threshold
    -tielo_pin  tielo_pin_name
    -tiehi_pin  tiehi_pin_name
```

Options

Switch Name	Description
`-liberty_file`	Liberty file with description of cells used in design. This is passed to ABC.
`-target`	Either `area` or `delay`. In area mode, the focus is area reduction, and timing may degrade. In delay mode, delay is likely reduced, but the area may increase.
`-slack_threshold`	Specifies a (setup) timing slack value below which timing paths need to be analyzed for restructuring.
`-depth_threshold`	Specifies the path depth above which a timing path would be considered for restructuring.
`-tielo_pin`	Tie cell pin that can drive constant zero. The format is `lib/cell/pin`.
`-tiehi_pin`	Tie cell pin that can drive constant one. The format is `lib/cell/pin`.

Example scripts

Example scripts on running `rmp` for a sample design of `gcd` as follows:

```
./test/gcd_restructure.tcl
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

Authors

- Sanjiv Mathur
- Ahmad El Rouby

License

BSD 3-Clause License. See *LICENSE* file.

5.2.10 Initialize Floorplan

This tool initializes floorplan constraints, die/core area, and makes tracks.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Initialize Floorplan

Do note that there are two ways of setting the floorplan dimensions. The user can either specify manually die/core area, or specify the utilization/aspect ratio.

Method 1: Automatic die size calculation

```
initialize_floorplan
  [-utilization util]
  [-aspect_ratio ratio]
  [-core_space space | {bottom top left right}]
  [-sites site_name]
```

Options

Switch Name	Description
`-utilization`	Percentage utilization. Allowed values are `double` in the range `(0-100]`.
`-aspect_ratio`	Ratio $\frac{height}{width}$. The default value is `1.0` and the allowed values are floats `[0, 1.0]`.
`-core_space`	Space around the core, default `0.0` microns. Allowed values are either one value for all margins or a set of four values, one for each margin. The order of the four values are: `{bottom top left right}`.
`-sites`	Tcl list of sites to make rows for (e.g. `{SITEXX, SITEYY}`)

Method 2: Set die/core area

```
initialize_floorplan
  [-die_area {llx lly urx ury}]
  [-core_area {llx lly urx ury}]
  [-sites site_name]
```

Options

Switch Name	Description
`-die_area`	Die area coordinates in microns (lower left x/y and upper right x/y coordinates).
`-core_area`	Core area coordinates in microns (lower left x/y and upper right x/y coordinates).
`-sites`	Tcl list of sites to make rows for (e.g. `{SITEXX, ...}`)

The die area and core area used to write ROWs can be specified explicitly with the `-die_area` and `-core_area` arguments. Alternatively, the die and core areas can be computed from the design size and utilization as shown below:

Example computation:

```
core_area = design_area / (utilization / 100)
core_width = sqrt(core_area / aspect_ratio)
core_height = core_width * aspect_ratio
core = ( core_space_left, core_space_bottom )
       ( core_space_left + core_width, core_space_bottom + core_height )
die =  ( 0, 0 )
       ( core_width + core_space_left + core_space_right,
         core_height + core_space_bottom + core_space_top )
```

Make Tracks

The `initialize_floorplan` command removes existing tracks.

Use the `make_tracks` command to add routing tracks to a floorplan.

```
make_tracks
    [layer]
    [-x_pitch x_pitch]
    [-y_pitch y_pitch]
```

(continues on next page)

(continued from previous page)

```
    [-x_offset x_offset]
    [-y_offset y_offset]
```

Options

Switch Name	Description
`layer`	Select layer name to make tracks for. Defaults to all layers.
`-x_pitch`, `-y_pitch`	If set, overrides the LEF technology x-/y- pitch. Use the same unit as in the LEF file.
`-x_offset`, `-y_offset`	If set, overrides the LEF technology x-/y- offset. Use the same unit as in the LEFT file.

Inserting tieoff cells

To insert tiecells:

```
insert_tiecells
    tie_pin
    [-prefix inst_prefix]
```

Options

Switch Name	Description
`tie_pin`	Indicates the master and port to use to tie off nets. For example, `LOGIC0_X1/Z` for the Nangate45 library, where `LOGIC0_X1` is the master and `Z` is the output port on the master.
`-prefix`	Used to control the prefix of the new tiecell names. This will default to `TIEOFF_`.

Useful developer functions

If you are a developer, you might find these useful. More details can be found in the source file or the *swig file*.

Command Name	Description
`microns_to_mfg_grid`	Convert microns to manufacturing grid DBU.

Example scripts

Example scripts on running `ifp` for a sample design of `mpd_top` are as follows:

```
./test/upf_test.tcl
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

License

BSD 3-Clause License. See *LICENSE* file.

5.2.11 Pin Placer

Place pins on the boundary of the die on the track grid to minimize net wirelengths. Pin placement also creates a metal shape for each pin using min-area rules.

For designs with unplaced cells, the net wirelength is computed considering the center of the die area as the unplaced cells' position.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Define Pin Shape Pattern

The `define_pin_shape_pattern` command defines a pin placement grid on the specified layer. This grid has positions inside the die area, not only at the edges of the die boundary.

```
define_pin_shape_pattern
    -layer layer
    -x_step x_step
    -y_step y_step
    -region {llx lly urx ury}
    -size {width height}
    [-pin_keepout dist]
```

Options

Switch Name	Description
`-layer`	The single top-most routing layer of the placement grid.
`-x_step`, `-y_step`	The distance (in microns) between each valid position on the grid in the x- and y-directions, respectively.
`-region`	The `{llx, lly, urx, ury}` region of the placement grid (in microns).
`-size`	The width and height (in microns) of the pins assigned to this grid. The centers of the pins are placed on the grid positions. Pins may have half of their shapes outside the defined region.
`-pin_keepout`	The boundary (in microns) around existing routing obstructions that the pins should avoid; this defaults to the `layer` minimum spacing.

Face-to-Face direct-bonding IOs

The `define_pin_shape_pattern` command can be used to place pins in any metal layer with the minimum allowed spacing to facilitate 3DIC integration of chips using face-to-face packaging technologies. These technologies include micro bumps and hybrid bonding for high density face-to-face interconnect.

Set IO Pin Constraint

The `set_io_pin_constraint` command sets region constraints for pins according to the pin direction or the pin name. This command can be called multiple times with different constraints.

You can use the `set_io_pin_constraint` command to restrict pins to the pin placement grid created with the `define_pin_shape_pattern` command.

It is possible to use the `-region`, `-group` and `-order` arguments together per `set_io_pin_constraint` call, but the `-mirrored_pins` argument should be called alone.

```
set_io_pin_constraint
    [-direction direction]
    [-pin_names names]
    [-region edge:interval]
    [-mirrored_pins names]
    [-group]
    [-order]
```

Options

Switch Name	Description
-direction	Pin direction (input, output, inout, or feedthrough).
-pin_names	List of names. Only one of (-direction, -pin_names) should be used in a single call for the set_io_pin_constraint command.
-region	Syntax is -region edge:interval. The edge values are (`top
-mirrored_pins	List of pins that sets pairs of pins that will be symmetrically placed in the vertical or the horizontal edges. The number of pins in this list **must be even**. For example, in set_io_pin_constraint -mirrored_pins {pin1 pin2 pin3 pin4 pin5 pin6}, the pins pin1 and pin2 will be placed symmetrically to each other. Same for pin3 and pin4, and for pin5 and pin6.
-group	Flag places together on the die boundary the pin list defined in -pin_names, similar to the -group_pins option on the place_pins command.
-order	Flag places the pins ordered in ascending x/y position and must be used only when -group is also used.

The edge values are (up, top, bottom, left, right), where up is the grid created by define_pin_shape_pattern. To restrict pins to the pin placement grid defined with define_pin_shape_pattern use:

- -region up:{llx lly urx ury} to restrict the pins into a specific region in the grid. The region is defined in microns.
- -region up:* to restrict the pins into the entire region of the grid.

The up option is only available when the pin placement grid is created with the define_pin_shape_pattern command.

Clear IO Pin Constraints

The clear_io_pin_constraints command clears all the previously-defined constraints and pin shape patterns created with set_io_pin_constraint or define_pin_shape_pattern.

```
clear_io_pin_constraints
```

Set Pin Length

The set_pin_length command defines the length of all vertical and horizontal pins.

```
set_pin_length
    [-hor_length h_length]
    [-ver_length v_length]
```

Options

Switch Name	Description
-hor_length	The length (in microns) of the horizontal pins.
-ver_length	The length (in microns) of the vertical pins.

Set Pin Extension

The set_pin_length_extension command defines the an extension of the length of all vertical and horizontal pins. Note that this command may generate pins partially outside the die area.

```
set_pin_length_extension
    [-hor_extension h_extension]
    [-ver_extension v_extension]
```

Options

Switch Name	Description
-hor_extension	The length (in microns) for the horizontal pins.
-ver_extension	The length (in microns) for the vertical pins.

Set Pin Thick Multiplier

The set_pin_thick_multiplier command defines a multiplier for the thickness of all vertical and horizontal pins.

```
set_pin_thick_multiplier
    [-hor_multiplier h_mult]
    [-ver_multiplier v_mult]
```

Options

Switch Name	Description
-hor_multiplier	The thickness multiplier for the horizontal pins.
-ver_multiplier	The thickness multiplier for the vertical pins.

Set Simulated Annealing Parameters

The set_simulated_annealing command defines the parameters for simulated annealing pin placement.

```
set_simulated_annealing
    [-temperature temperature]
    [-max_iterations iter]
    [-perturb_per_iter perturbs]
    [-alpha alpha]
```

Options

Switch Name	Description
-temperature	Temperature parameter. The default value is 1.0, and the allowed values are floats [0, MAX_FLOAT].
-max_iterations	The maximum number of iterations. The default value is 2000, and the allowed values are integers [0, MAX_INT].
-perturb_per_iter	The number of perturbations per iteration. The default value is 0, and the allowed values are integers [0, MAX_INT].
-alpha	The temperature decay factor. The default value is 0.985, and the allowed values are floats (0, 1].

Place Individual Pin

The `place_pin` command places a specific pin in the specified location with the specified size. It is recommended that individual pins be placed before the `place_pins` command, as the routing tracks occupied by these individual pins will be blocked, preventing overlaps.

To place an individual pin:

```
place_pin
    -pin_name pin_name
    -layer layer
    -location {x y}
    [-pin_size {width height}]
    [-force_to_die_boundary]
```

Options

Switch Name	Description
-pin_name	The name of a pin of the design.
-layer	The routing layer where the pin is placed.
-location	The center of the pin (in microns).
-pin_size	Tthe width and height of the pin (in microns).
-force_to_die_boundary	When this flag is enabled, the pin will be snapped to the nearest routing track, next to the die boundary.

Place All Pins

Use the following command to perform pin placement:

```
place_pins
    -hor_layers h_layers
    -ver_layers v_layers
    [-random_seed seed]
    [-random]
    [-corner_avoidance length]
    [-min_distance distance]
```

(continues on next page)

(continued from previous page)

```
    [-min_distance_in_tracks]
    [-exclude region]
    [-group_pins pin_list]
    [-annealing]
    [-write_pin_placement file_name]
```

Options

Switch Name	Description
-hor_layers	The layers to create the metal shapes of pins placed in horizontal tracks. It can be a single layer or a list of layer names.
-ver_layers	The layers to create the metal shapes of pins placed in vertical tracks. It can be a single layer or a list of layer names.
-corner_avoidance	The distance (in microns) from each corner within which pin placement should be avoided.
-min_distance	The minimum distance between pins on the die boundary. This distance can be in microns (default) or in number of tracks between each pin.
-min_distance_in_tracks	Flag that allows setting the min distance in number of tracks instead of microns.
-exclude	A region where pins cannot be placed. Either `top
-group_pins	A list of pins to be placed together on the die boundary.
-annealing	Flag to enable simulated annealing pin placement.
-write_pin_placement	A file with the pin placement generated in the format of multiple calls for the place_pin command.

The exclude option syntax is -exclude edge:interval. The edge values are (top|bottom|left|right). The interval can be the whole edge, with the * value, or a range of values. For example, in place_pins -hor_layers metal2 -ver_layers metal3 -exclude top:* -exclude right:15-60.5 -exclude left:*-50 three intervals are excluded: the whole top edge, the right edge from 15 microns to 60.5 microns, and the left edge from its beginning to 50 microns.

Developer Arguments

Switch Name	Description
-random_seed	Specify the seed for random operations.
-random	When this flag is enabled, the pin placement is random.

Write Pin Placement

Use the following command to write a file with the pin placement in the format of multiple calls for the place_pin command:

```
write_pin_placement file_name
```

Options

Switch Name	Description
file_name	The name of the file with the pin placement.

Useful Developer Commands

If you are a developer, you might find these useful. More details can be found in the source file or the swig file.

Command Name	Description
parse_edge	Parse edge (top/bottom/left/right).
parse_direction	Parse direction.
parse_excludes_arg	Parse excluded arguments.
parse_group_pins_arg	Parse group pins arguments.
parse_layer_name	Parse layer name.
parse_pin_names	Parse pin names.
get_edge_extreme	Get extremes of edge.
exclude_intervals	Set exclude interval.
add_pins_to_constraint	Add pins to constrained region.
add_pins_to_top_layer	Add pins to top layer.

Configure Simulated Annealing Solver

The `set_simulated_annealing` command defines the parameters of the Simulated Annealing solver.

```
set_simulated_annealing [-temperature temperature]
                        [-max_iterations iters]
                        [-perturb_per_iter perturbs]
                        [-alpha alpha]
```

- The `-temperature` sets the initial temperature of the Simulated Annealing solver.
- The `-max_iterations` sets the number of iterations performed by the Simulated Annealing solver.
- The `-perturb_per_iter` sets the number of perturbations performed at each iteration.
- The `-alpha` sets the reduction factor of the temperature at each iteration.

Example scripts

Example scripts of `ppl` running on a sample design of `gcd` as follows:

```
./test/gcd.tcl
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

References

- This code depends on Munkres.

FAQs

Check out GitHub discussion about this tool.

License

BSD 3-Clause License. See LICENSE file.

5.2.12 Chip-level Connections

The chip-level connections module in OpenROAD (`pad`) is based on the open-source tool ICeWall. In this utility, either place an IO ring around the boundary of the chip and connect with either wirebond pads or a bump array.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Placing Terminals

In the case where the bond pads are integrated into the padcell, the IO terminals need to be placed. To place a terminals on the padring

```
place_io_terminals inst_pins
```

Options

Switch Name	Description
inst_pins	Instance pins to place the terminals on.

Examples

```
place_io_terminals u_*/PAD
place_io_terminals u_*/VDD
```

Defining a Bump Array

To define a bump array.

```
make_io_bump_array
    -bump master
    -origin {x y}
    -rows rows
    -columns columns
    -pitch {x y}
    [-prefix prefix]
```

Options

Switch Name	Description
-bump	Name of the bump master.
-origin	Origin of the array.
-rows	Number of rows to create.
-columns	Number of columns to create.
-pitch	Pitch of the array.
-prefix	Name prefix for the bump array. The default value is BUMP_.
Example usage:	

```
make_io_bump_array -bump BUMP -origin "200 200" -rows 14 -columns 14 -pitch "200 200"
```

Removing Entire Bump Array

To remove a bump array.

```
remove_io_bump_array -bump master
```

Options

Switch Name	Description
-bump	Name of the bump master.

Example usage:

```
remove_io_bump_array -bump BUMP
```

Removing a Single Bump Instance

To remove a single bump instance.

```
remove_io_bump instance_name
```

Options

Switch Name	Description
instance_name	Name of the bump.

Assigning a Net to a Bump

To assign a net to a bump.

```
assign_io_bump
    -net net
    [-terminal iterm]
    [-dont_route]
    instance
```

Options

Switch Name	Description
-net	Net to connect to.
-terminal	Instance terminal to route to.
-dont_route	Flag to indicate that this bump should not be routed, only perform assignment.
instance	Name of the bump.

Example usage:

```
assign_io_bump -net p_ddr_addr_9_o BUMP_6_0
assign_io_bump -net p_ddr_addr_8_o BUMP_6_2
assign_io_bump -net DVSS BUMP_6_4
assign_io_bump -net DVDD BUMP_7_3
assign_io_bump -net DVDD -terminal u_dvdd/DVDD BUMP_8_3
```

(continues on next page)

(continued from previous page)

```
assign_io_bump -net p_ddr_addr_7_o BUMP_7_1
assign_io_bump -net p_ddr_addr_6_o BUMP_7_0
```

Define IO Rows

Define an IO site for the pads to be placed into.

```
make_io_sites
    -horizontal_site site
    -vertical_site site
    -corner_site site
    -offset offset
    [-rotation_horizontal rotation]
    [-rotation_vertical rotation]
    [-rotation_corner rotation]
    [-ring_index index]
```

Options

Switch Name	Description
-horizontal_site	Name of the site for the horizontal pads (east and west).
-vertical_site	Name of the site for the vertical pads (north and south).
-corner_site	Name of the site for the corner cells.
-offset	Offset from the die edge to place the rows.
-rotation_horizontal	Rotation to apply to the horizontal sites to ensure pads are placed correctly. The default value is R0.
-rotation_vertical	Rotation to apply to the vertical sites to ensure pads are placed correctly. The default value is R0.
-rotation_corner	Rotation to apply to the corner sites to ensure pads are placed correctly. The default value is R0.
-ring_index	Used to specify the index of the ring in case of multiple rings.

Example usage:

```
make_io_sites -horizontal_site IOSITE_H -vertical_site IOSITE_V -corner_site IOSITE_C -
↪offset 35
make_io_sites -horizontal_site IOSITE_H -vertical_site IOSITE_V -corner_site IOSITE_C -
↪offset 35 -rotation_horizontal R180
```

Remove IO Rows

When the padring is complete, the following command can remove the IO rows to avoid causing confusion with the other tools.

```
remove_io_rows
```

Placing Corners

To place the corner cells

```
place_corners
    master
    [-ring_index index]
```

Options

Switch Name	Description
master	Name of the master for the corners.
-ring_index	Used to specify the index of the ring in case of multiple rings.

Example usage:

```
place_corners sky130_fd_io__corner_bus_overlay
```

Placing Pads

To place a pad into the pad ring.

```
place_pad
    -row row_name
    -location offset
    -mirror
    [-master master]
    name
```

Options

Switch Name	Description
-row	Name of the row to place the pad into, examples include: IO_NORTH, IO_SOUTH, IO_WEST, IO_EAST, IO_NORTH_0, IO_NORTH_1.
-location	Offset from the bottom left chip edge to place the pad at.
-mirror	Specifies if the pad should be mirrored.
-master	Name of the instance master if the instance needs to be created.
name	Name of the instance.

Example usage:

```
place_pad -row IO_SOUTH -location 280.0 {u_clk.u_in}
place_pad -row IO_SOUTH -location 360.0 -mirror {u_reset.u_in}
place_pad -master sky130_fd_io__top_ground_hvc_wpad -row IO_SOUTH -location 439.5 {u_vzz_
↪0}
place_pad -master sky130_fd_io__top_power_hvc_wpad -row IO_SOUTH -location 517.5 {u_v18_
↪0}
```

Placing IO Filler Cells

To place the IO filler cells.

```
place_io_fill
    -row row_name
    [-permit_overlaps masters]
    masters
```

Options

Switch Name	Description
-row	Name of the row to place the pad into, examples include: IO_NORTH, IO_SOUTH, IO_WEST, IO_EAST, IO_NORTH_0, IO_NORTH_1.
-permit_overlaps	Names of the masters for the IO filler cells that allow for overlapping.
masters	Names of the masters for the IO filler cells.

Example usage:

```
place_io_fill -row IO_NORTH s8iom0s8_com_bus_slice_10um s8iom0s8_com_bus_slice_5um␣
↪s8iom0s8_com_bus_slice_1um
place_io_fill -row IO_SOUTH s8iom0s8_com_bus_slice_10um s8iom0s8_com_bus_slice_5um␣
↪s8iom0s8_com_bus_slice_1um
place_io_fill -row IO_WEST s8iom0s8_com_bus_slice_10um s8iom0s8_com_bus_slice_5um␣
↪s8iom0s8_com_bus_slice_1um
place_io_fill -row IO_EAST s8iom0s8_com_bus_slice_10um s8iom0s8_com_bus_slice_5um␣
↪s8iom0s8_com_bus_slice_1um
```

Connecting Ring Signals

Once the ring is complete, use the following command to connect the ring signals.

```
connect_by_abutment
```

Placing Wirebond Pads

To place the wirebond pads over the IO cells.

```
place_bondpad
    -bond master
    [-offset {x y}]
    [-rotation rotation]
    io_instances
```

Options

Switch Name	Description
-bond	Name of the bondpad master.
-offset	Offset to place the bondpad at with respect to the io instance.
-rotation	Rotation of the bondpad.
io_instances	Names of the instances to add bond pads to.

Example usage:

```
place_bondpad -bond PAD IO_*
```

Creating False IO Sites

If the library does not contain sites for the IO cells, the following command can be used to add them. This should not be used unless the sites are not in the library.

```
make_fake_io_site
    -name name
    -width width
    -height height
```

Options

Switch Name	Description
-name	Name of the site.
-width	Width of the site (in microns).
-height	Height of the site (in microns).

Example usage:

```
make_fake_io_site -name IO_HSITE -width 1 -height 204
make_fake_io_site -name IO_VSITE -width 1 -height 200
make_fake_io_site -name IO_CSITE -width 200 -height 204
```

Redistribution Layer Routing

To route the RDL for the bump arrays.

```
rdl_route
    -layer layer
    [-bump_via access_via]
    [-pad_via access_via]
    [-width width]
    [-spacing spacing]
    [-turn_penalty penalty]
    [-allow45]
    nets
```

Options

Switch Name	Description
`-layer`	Layer to route on.
`-bump_via`	Via to use to to connect the bump to the routing layer.
`-pad_via`	Via to use to to connect the pad cell to the routing layer.
`-width`	Width of the routing. Defaults to minimum width for each respective layer.
`-spacing`	Spacing of the routing. Defaults to minimum spacing for each respective layer.
`-turn_penalty`	Scaling factor to apply to discurage turning to allow for straighter routes. The default value is `2.0`, and the allowed values are floats.
`-allow45`	Specifies that 45 degree routing is permitted.
`nets`	Nets to route.

Useful Developer Commands

If you are a developer, you might find these useful. More details can be found in the source file or the swig file.

Command Name	Description
`find_site`	Find site given site name.
`find_master`	Find master given master name.
`find_instance`	Find instance given instance name.
`find_net`	Find net given net name.
`assert_required`	Assert argument that is required for `cmd`
`connect_iterm`	Connect instance terminals. Required inputs are: `inst_name`, `iterm_name`, `net_name`.
`convert_tcl`	These functions read from $ICeWall::library parameters to generate a standalone Tcl script.

Example Scripts

Example scripts for running ICeWall functions can be found in `./test`.

```
./test/assign_bumps.tcl
./test/bump_array_make.tcl
./test/bump_array_remove.tcl
./test/bump_array_remove_single.tcl
./test/connect_by_abutment.tcl
./test/make_io_sites.tcl
./test/place_bondpad.tcl
./test/place_bondpad_stagger.tcl
./test/place_pad.tcl
./test/rdl_route.tcl
./test/rdl_route_45.tcl
./test/rdl_route_assignments.tcl
```

Regression Tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

License

BSD 3-Clause License. See LICENSE file.

5.2.13 Macro Placement

The macro placement module in OpenROAD (`mpl`) is based on TritonMacroPlacer, an open-source ParquetFP-based macro cell placer. The macro placer places macros/blocks honoring halos, channels and cell row "snapping". Run `global_placement` before macro placement.

Approximately $\left\lceil [\frac{numMacros}{3}]^{1.5} \right\rceil$ quadrisections of the initial placed mixed-size layout are explored and packed using ParquetFP-based annealing. The best resulting floorplan according to a heuristic evaluation function is kept.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Macro Placement

```
macro_placement
    [-halo {halo_x halo_y}]
    [-channel {channel_x channel_y}]
    [-fence_region {lx ly ux uy}]
    [-snap_layer snap_layer_number]
    [-style corner_wax_wl|corner_min_wl]
```

Options

Switch Name	Description
`-halo`	Horizontal and vertical halo around macros (microns).
`-channel`	Horizontal and vertical channel width between macros (microns).
`-fence_region`	Restrict macro placements to a region (microns). Defaults to the core area.
`-snap_layer_number`	Snap macro origins to this routing layer track.
`-style`	Placement style, to choose either `corner_max_wl` or `corner_min_wl`. The default value is `corner_max_wl`.

For placement style, `corner_max_wl` means that choosing the partitions that maximise the wirelength of connections between the macros to force them to the corners. Vice versa for `corner_min_wl`.

Macros will be placed with $max(halo * 2, channel)$ spacing between macros, and between macros and the fence/die boundary. If no solutions are found, try reducing the channel/halo.

Useful developer functions

If you are a developer, you might find these useful. More details can be found in the source file or the swig file.

Command Name	Description
`macro_placement_debug`	Macro placement debugging. Note that GUI must be present for this command, otherwise a segfault will occur.

Example scripts

Example scripts demonstrating how to run TritonMacroPlace on a sample design of `east_west` as follows:

```
./test/east_west.tcl
./test/east_west1.tcl
./test/east_west2.tcl
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

License

BSD 3-Clause License. See LICENSE file.

5.2.14 Hierarchical Macro Placement

A hierarchical automatic macro placer for large-scale complex IP blocks, "Hier-RTLMP". This tool builds on the existing RTLMP (`mpl`) framework, adopting a multilevel physical planning approach that exploits the hierarchy and data flow inherent in the design RTL.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Hier-RTLMP algorithm

```
rtl_macro_placer
    [-halo_width halo_width]
    [-min_num_macro min_num_macro]
    [-max_num_inst max_num_inst]
    [-min_num_inst min_num_inst]
    [-tolerance tolerance]
    [-max_num_level max_num_level]
    [-coarsening_ratio coarsening_ratio]
    [-num_bundled_ios num_bundled_ios]
    [-large_net_threshold large_net_threshold]
    [-signature_net_threshold signature_net_threshold]
    [-halo_width halo_width]
    [-fence_lx fence_lx]
    [-fence_ly fence_ly]
    [-fence_ux fence_ux]
    [-fence_uy fence_uy]
    [-area_weight area_weight]
    [-outline_weight outline_weight]
    [-wirelength_weight wirelength_weight]
    [-guidance_weight guidance_weight]
    [-fence_weight fence_weight]
    [-boundary_weight boundary_weight]
    [-notch_weight notch_weight]
    [-macro_blockage_weight macro_blockage_weight]
    [-target_util target_util]
    [-target_dead_space target_dead_space]
    [-min_ar min_ar]
    [-snap_layer snap_layer]
    [-bus_planning_flag bus_planning_flag]
    [-report_directory report_directory]
    [-write_macro_placement file_name]
```

Generic Parameters

Switch Name	Description
-max_num_macro, -min_num_macro	Maximum/minimum number of macros in a cluster. The default value is 0 for both, and the allowed values are integers [0, MAX_INT].
-max_num_inst, -min_num_inst	Maximum/minimum number of standard cells in a cluster. The default value is 0 for both, and the allowed values are integers [0, MAX_INT].
-tolerance	Add a margin to the minimum and maximum number of macros/std cells in a cluster. For min, we multiply by (1 - tol), and for the max (1 + tol). This is to improve the robustness of hierarchical clustering. The allowed values are floats [0, 1), and the default value is 0.1.
-max_num_level	Maximum depth of physical hierarchical tree. The default value is 2, and the allowed values are integers [0, MAX_INT].
-coarsening_ratio	The larger the coarsening_ratio, the faster the convergence process. The allowed values are floats, and the default value is 10.0.
-num_bundled_ios	Specifies the number of bundled pins for the left, right, top, and bottom boundaries. The default value is 3, and the allowed values are integers [0, MAX_INT].
-large_net_threshold	Ignore nets with many connections during clustering, such as global nets. The default value is 50, and the allowed values are integers [0, MAX_INT].
-signature_net_threshold	Minimum number of connections between two clusters to be identified as connected. The default value is 50, and the allowed values are integers [0, MAX_INT].
-halo_width	iHorizontal/vertical halo around macros (microns). The allowed values are floats, and the default value is 0.0.
-fence_lx, -fence_ly, -fence_ux, -fence_uy	Defines the global fence bounding box coordinates. The default values are the core area coordinates).
-target_util	Specifies the target utilization of MixedCluster and has higher priority than target_dead_space. The allowed values are floats, and the default value is 0.25.
-target_dead_space	Specifies the target dead space percentage, which influences the utilization of StandardCellCluster. The allowed values are floats, and the default value is 0.05.
-min_ar	Specifies the minimum aspect ratio a, or the ratio of its width to height of a StandardCellCluster from $[a, \frac{1}{a}]$. The allowed values are floats, and the default value is 0.33.
-snap_layer	Snap macro origins to this routing layer track. The default value is 4, and the allowed values are integers [1, MAX_LAYER]).
-bus_planning_flag	Flag to enable bus planning. The recommendation is to turn on bus planning for SKY130 and off for NanGate45/ASAP7. The default value is disabled.
-report_directory	Save reports to this directory.
-write_macro_placement	Generates a file with the macro placement in the format of multiple calls for the place_macro command.

Simulated Annealing Weight parameters

Do note that while action probabilities are normalized to 1.0, the weights are not necessarily normalized.

Switch Name	Description
-area_weight	Weight for the area of current floorplan. The allowed values are floats, and the default value is 0.1.
-outline_weight	Weight for violating the fixed outline constraint, meaning that all clusters should be placed within the shape of their parent cluster. The allowed values are floats, and the default value is 100.0.
-wirelength_weight	Weight for half-perimeter wirelength. The allowed values are floats, and the default value is 100.0.
-guidance_weight	Weight for guidance cost or clusters being placed near specified regions if users provide such constraints. The allowed values are floats, and the default value is 10.0.
-fence_weight	Weight for fence cost, or how far the macro is from zero fence violation. The allowed values are floats, and the default value is 10.0.
-boundary_weight	Weight for the boundary, or how far the hard macro clusters are from boundaries. Note that mixed macro clusters are not pushed, thus not considered in this cost. The allowed values are floats, and the default value is 50.0.
-notch_weight	Weight for the notch, or the existence of dead space that cannot be used for placement & routing. Note that this cost applies only to hard macro clusters. The allowed values are floats, and the default value is 10.0.
-macro_blockage_weight	Weight for macro blockage, or the overlapping instances of the macro. The allowed values are floats, and the default value is 10.0.

Write Macro Placement

Command to write a file with the macro placement in the format of multiple calls for the `place_macro` command:

```
write_macro_placement file_name
```

Place Macro

Command for placement of one specific macro.

```
place_macro
    -macro_name macro_name
    -location {x y}
    [-orientation orientation]
```

Options

Switch Name	Description
-macro_name	The name of a macro of the design.
-location	The lower left corner of the macro in microns.
-orientation	The orientation according to odb. If nothing is specified, defaults to R0. We only allow R0, MY, MX and R180.

Example scripts

Example of a script demonstrating how to run `mpl2` on a sample design of `bp_fe_top` as follows:

```
./test/bp_fe_top.tcl
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

References

1. A. B. Kahng, R. Varadarajan and Z. Wang, "RTL-MP: Toward Practical, Human-Quality Chip Planning and Macro Placement", (.pdf), Proc. ACM/IEEE Intl. Symp. on Physical Design, 2022, pp. 3-11.
2. A. B. Kahng, R. Varadarajan and Z. Wang, "Hier-RTLMP: A hierarchical automatic macro placer for large-scale complex IP blocks.", (.pdf), arXiv preprint arXiv:2304.11761, 2023.

FAQs

Check out GitHub discussion about this tool.

License

5.2.15 Tapcell

Tapcell and endcap insertion.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Add Tapcell/Endcap

```
tapcell
    [-tapcell_master tapcell_master]
    [-endcap_master endcap_master]
    [-distance dist]
    [-halo_width_x halo_x]
    [-halo_width_y halo_y]
    [-tap_nwin2_master tap_nwin2_master]
    [-tap_nwin3_master tap_nwin3_master]
    [-tap_nwout2_master tap_nwout2_master]
    [-tap_nwout3_master tap_nwout3_master]
    [-tap_nwintie_master tap_nwintie_master]
    [-tap_nwouttie_master tap_nwouttie_master]
    [-cnrcap_nwin_master cnrcap_nwin_master]
    [-cnrcap_nwout_master cnrcap_nwout_master]
    [-incnrcap_nwin_master incnrcap_nwin_master]
    [-incnrcap_nwout_master incnrcap_nwout_master]
    [-tap_prefix tap_prefix]
    [-endcap_prefix endcap_prefix]
```

Options

Switch Name	Description
-tapcell_master	Master used as a tapcell.
-endcap_master	Master used as an endcap.
-distance	Distance (in microns) between each tapcell in the checkerboard.
-halo_width_x	Horizontal halo size (in microns) around macros during cut rows.
-halo_width_y	Vertical halo size (in microns) around macros during cut rows.
-tap_nwintie_master	Master cell placed at the top and bottom of
-tap_nwin2_master	Master cell placed at the top and bottom of macros and the core area according the row orientation. This master should be smaller than tap_nwintie_master
-tap_nwin3_master	Master cell placed at the top and bottom of macros and the core area according the row orientation. This master should be smaller than tap_nwin2_master.
-tap_nwouttie_master	Master cell placed at the top and bottom of macros and the core area according the row orientation.
-tap_nwout2_master	Master cell placed at the top and bottom of macros and the core area according the row orientation. This master should be smaller than tap_nwouttie_master.
-tap_nwout3_master	Master cell placed at the top and bottom of macros and the core area according the row orientation
-incnrcap_nwin_master	Master cell placed at the corners of macros, according the row orientation.
-incnrcap_nwout_master	Master cell placed at the corners of macros, according the row orientation.
-cnrcap_nwin_master	Macro cell placed at the corners the core area according the row orientation.
-cnrcap_nwout_master	Macro cell placed at the corners the core area according the row orientation.
-tap_prefix	Prefix for the tapcell instances. The default value is TAP_.
-endcap_prefix	Prefix for the endcaps instances. The default value is PHY_.

The figures below show two examples of tapcell insertion. When only the -tapcell_master and -endcap_master masters are given, the tapcell placement is similar to Figure 1. When the remaining masters are give, the tapcell placement is similar to Figure 2.

Figure 1: Tapcell insertion representation	Figure 2: Tapcell insertion around macro representation

Only cutting rows

```
cut_rows
    [-endcap_master endcap_master]
    [-halo_width_x halo_x]
    [-halo_width_y halo_y]
```

Options

Switch Name	Description
-endcap_master	Master used as an endcap.
-halo_width_x	Horizontal halo size (in microns) around macros during cut rows.
-halo_width_y	Vertical halo size (in microns) around macros during cut rows.

Only adding boundary/endcap cells

Place endcaps into the design, the naming for the arguments to `place_endcaps` is based on the LEF58 `CLASS` specification foe endcaps.

```
place_endcaps
    [-corner master]
    [-edge_corner master]
    [-endcap masters]
    [-endcap_horizontal masters]
    [-endcap_vertical master]
    [-prefix prefix]
    [-left_top_corner master]
    [-right_top_corner master]
    [-left_bottom_corner master]
    [-right_bottom_corner master]
    [-left_top_edge master]
    [-right_top_edge master]
    [-left_bottom_edge master]
    [-right_bottom_edge master]
    [-left_edge master]
    [-right_edge master]
    [-top_edge masters]
    [-bottom_edge masters]
```

Options

Switch Name	Description
-prefix	Prefix to use for the boundary cells. Defaults to "PHY_".
-corner	Master for the corner cells on the outer corners.
-edge_corner	Master for the corner cells on the inner corners.
-endcap	Master used as an endcap.
-endcap_horizontal	List of masters for the top and bottom row endcaps. (overrides -endcap).
-endcap_vertical	Master for the left and right row endcaps. (overrides -endcap).
-left_top_corner	Master for the corner cells on the outer top left corner. (overrides -corner).
-right_top_corner	Master for the corner cells on the outer top right corner. (overrides -corner).
-left_bottom_corner	Master for the corner cells on the outer bottom left corner. (overrides -corner).
-right_bottom_corner	Master for the corner cells on the outer bottom right corner. (overrides -corner).
-left_top_edge	Master for the corner cells on the inner top left corner. (overrides -edge_corner).
-right_top_edge	Master for the corner cells on the inner top right corner. (overrides -edge_corner).
-left_bottom_edge	Master for the corner cells on the inner bottom left corner. (overrides -edge_corner).
-right_bottom_edge	Master for the corner cells on the inner bottom right corner. (overrides -edge_corner).
-left_edge	Master for the left row endcaps. (overrides -endcap_vertical).
-right_edge	Master for the right row endcaps. (overrides -endcap_vertical).
-top_edge	List of masters for the top row endcaps. (overrides -endcap_horizontal).
-bottom_edge	List of masters for the bottom row endcaps. (overrides -endcap_horizontal).

Only adding tapcells cells

```
place_tapcells
    -master tapcell_master
    -distance dist
```

Options

Switch Name	Description
-master	Master to use for the tapcells.
-distance	Distance between tapcells.

Remove Tapcells/Endcaps

```
tapcell_ripup
    -tap_prefix tap_prefix
    -endcap_prefix endcap_prefix
```

Options

Switch Name	Description
`-tap_prefix`	Remove tapcells with said prefix.
`-endcap_prefix`	Remove endcaps with said prefix.

Example scripts

You can find script examples for both 45nm and 14nm in `./etc/scripts`

```
./etc/scripts/example_14nm.tcl
./etc/scripts/example_45nm.tcl
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

License

BSD 3-Clause License. See *LICENSE* file.

5.2.16 PDNGEN

This utility aims to simplify the process of adding a power grid into a floorplan. The aim is to specify a small set of power grid policies to be applied to the design, such as layers to use, stripe width and spacing, then have the utility generate the actual metal straps. Grid policies can be defined over the stdcell area, and over areas occupied by macros.

See also:

To work with UPF files, refer to *Read UPF Utility*.

Commands

Define Power Switch Cell

Define a power switch cell that will be inserted into a power grid

```
define_power_switch_cell -name <name> \
                         -control <control_pin_name> \
                         [-acknowledge <acknowledge_pin_name>] \
                         -switched_power <switched_power_pin> \
                         -power <unswitched_power_pin> \
                         -ground <ground_pin>
```

Options

Switch Name	Description
-name	The name of the power switch cell.
-control	The name of the power control port of the power switch cell.
-acknowledge	Defines the name of the output control signal of the power control switch if it has one.
-switched_power	Defines the name of the pin that outputs the switched power net
-power	Defines the name of the pin that connects to the unswitched power net.
-ground	Defines the name of the pin that connects to the ground net.

Examples

```
define_power_switch_cell -name POWER_SWITCH -control SLEEP -switched_power VDD -power␣
↪VDDG -ground VSS
```

Define voltage domains

Defines a named voltage domain with the names of the power and ground nets for a region.

The -region argument specifies the name of a region of the design. This region must already exist in the floorplan before referencing it with the set_voltage_domain command. If the -region argument is not supplied then region is the entire extent of the design.

The -name argument is used to define a name for the voltage domain that can be used in the `define_pdn_grid` command The -power and -ground arguments are used to define the names of the nets to be use for power and ground respectively within this voltage domain.

If the voltage domain is a switched power domain, then the name of the swiched power net must be specified with the -switched_power option.

```
set_voltage_domain [-name name] \
                   -power power_net \
                   -ground ground_net \
                   [-region region_name] \
                   [-secondary_power secondary_power_net] \
                   [-switched_power <switched_power_net>]
```

Options

Switch Name	Description
-name	Defines the name of the voltage domain, default is “Core” or region name if provided
-power	Specifies the name of the power net for this voltage domain
-ground	Specifies the name of the ground net for this voltage domain
-region	Specifies a region of the design occupied by this voltage domain
-secondary_power	Specifies the name of the secondary power net for this voltage domain
-switched_power	Specifies the name of the switched power net for switched power domains,

Examples

```
set_voltage_domain -power VDD -ground VSS
set_voltage_domain -name TEMP_ANALOG -region TEMP_ANALOG -power VIN -ground VSS
set_voltage_domain -region test_domain -power VDD -ground VSS -secondary_power VREG
```

Define power grids

Define the rules to describe a power grid pattern to be placed in the design.

```
define_pdn_grid [-name <name>] \
                [-pins <list_of_pin_layers>] \
                [-starts_with (POWER|GROUND)] \
                [-voltage_domain <list_of_domain_names>] \
                [-starts_with (POWER|GROUND)] \
                [-obstructions <list_of_layers>]
```

Options

Switch Name	Description
-name	Defines a name to use when referring to this grid definition.
-voltage_domain	Defines the name of the voltage domain for this grid. (Default: Last domain created)
-pins	Defines a list of layers which where the power straps will be promoted to block pins.
-starts_with	Specifies whether the first strap placed will be POWER or GROUND (Default: GROUND)
-obstructions	Specify the layers to add routing blockages, in order to avoid DRC violations

Examples

```
define_pdn_grid -name main_grid -pins {metal7} -voltage_domain {CORE TEMP_ANALOG}
```

Define a macro power grid

```
define_pdn_grid -macro \
                [-name name] \
                [-grid_over_pg_pins|-grid_over_boundary] \
                [-orient <list_of_valid_orientations>] \
                [-instances <list_of_instances] \
                [-cells <list_of_cells>] \
                [-default] \
                [-halo <list_of_halo_values>] \
                [-voltage_domain <list_of_domain_names>] \
                [-starts_with (POWER|GROUND)] \
                [-obstructions <list_of_layers>] \
                [-bump]
```

Options

Switch Name	Description
-macro	Defines the type of grid being added as a macro.
-name	Defines a name to use when referring to this grid definition.
-voltage_domain	Defines the name of the voltage domain for this grid. (Default: Last domain created)
-starts_with	Specifies whether the first strap placed will be POWER or GROUND (Default: GROUND)
-grid_over_boundary	Place the power grid over the entire macro.
-grid_over_pg_pins	Place the power grid over the power ground pins of the macro. (Default)
-instances	For a macro, defines a set of valid instances. Macros with a matching instance name will use this grid specification.
-cells	For a macro, defines a set of valid cells. Macros which are instances of one of these cells will use this grid specification.
-default	For a macro, specifies this is a default grid that can be overwritten.
-orient	For a macro, defines a set of valid orientations. LEF orientations (N, FN, S, FS, E, FE, W and FW) can be used as well as standard geometry orientations (R0, R90, R180, R270, MX, MY, MXR90 and MYR90). Macros with one of the valid orientations will use this grid specification.
-halo	Specifies the default minimum separation of selected macros from other cells in the design. This is only used if the macro does not define halo values in the LEF description. If 1 value is specified it will be used on all 4 sides, if two values are specified, the first will be applied to left/right sides and the second will be applied to top/bottom sides, if 4 values are specified, then they are applied to left, bottom, right and top sides respectively. (Default: 0)
-obstructions	Specifies the layers to add routing blockages, in order to avoid DRC violations
-bump	Flag to indicate this is a grid for a cover cell

Examples

```
define_pdn_grid -macro -name ram          -orient {R0 R180 MX MY} -grid_over_pg_pins  -
↪starts_with POWER -pin_direction vertical
define_pdn_grid -macro -name rotated_rams -orient {E FE W FW}     -grid_over_boundary -
↪starts_with POWER -pin_direction horizontal
```

Define a grid for an existing routing

```
define_pdn_grid [-name <name>] \
                -existing \
                [-obstructions <list_of_layers>]
```

Options

Switch Name	Description
-name	Defines a name to use when referring to this grid definition. Defaults to existing_grid
-obstructions	Specify the layers to add routing blockages, in order to avoid DRC violations

Examples

```
define_pdn_grid -name main_grid -existing
```

Power switch insertion

```
define_pdn_grid [-name <name>] \
                [-switch_cell <power_switch_cell_name> ] \
                [-power_control <power_constrol_signal_name>] \
                [-power_control_network (STAR|DAISY)]
```

Options

Switch Name	Description
-switch_cell	Defines the name of the coarse grain power switch cell to be used for this grid.
-power_control	Defines the name of the power control signal used to control the switching of the inserted power switches.
-power_control_network	Defines the structure of the power control signal network. Choose from STAR, or DAISY

The -switch_cell argument is used to specify the name of a coarse-grain power switch cell that is to be inserted whereever the stdcell rail connects to the rest of the power grid. The mesh layers are associated with the unswitched power net of the voltage domain, whereas the stdcell rail is associated with the switched power net of the voltage domain. The placement of a power switch cell connects the unswitched power mesh to the switched power rail through a power switch defined by the define_power_switch_cell command.

The -power_control argument specifies the name of the power control signal that must be connected to the inserted power control cells.

The -power_control_network argument specifies how the power control signal is to be connected to the power switches. If STAR is specified, then the network is wired as a high-fanout net with the power control signal driving the power control pin on every power switch. If DAISY is specified then the power switches are connected in a daisy-chain configuration - note, this requires that the power swich defined by the define_power_switch_cell command defines an acknowledge pin for the switch.

Add straps / stripes

Defines a pattern of power and ground stripes in a single layer to be added to a power grid.

```
add_pdn_stripe [-grid grid_name] \
               -layer layer_name \
               -width width_value \
               [-pitch pitch_value] \
               [-spacing spacing_value] \
               [-offset offset_value] \
               [-starts_with (POWER|GROUND)] \
               [-followpins] \
               [-extend_to_boundary] \
               [-extend_to_core_ring] \
               [-snap_to_grid] \
               [-number_of_straps count] \
               [-nets list_of_nets]
```

Options

Switch Name	Description
-grid	Specifies the grid to which this stripe definition will be added. (Default: Last grid defined by define_pdn_grid)
-layer	Specifies the name of the layer for these stripes
-width	Value for the width of stripe
-pitch	Value for the distance between each power/ground pair
-spacing	Optional specification of the spacing between power/ground pairs within a single pitch. (Default: pitch / 2)
-offset	Value for the offset of the stripe from the lower left corner of the design core area.
-starts_with	Specifies whether the first strap placed will be POWER or GROUND (Default: grid setting)
-followpins	Indicates that the stripe forms part of the stdcell rails, pitch and spacing are dictated by the stdcell rows, the -width is not needed if it can be determined from the cells
-extend_to_boundary	Extend the stripes to the boundary of the grid
-snap_to_grid	Snap the stripes to the defined routing grid
-number_of_straps	Number of power/ground pairs to add
-nets	Limit straps to just this list of nets

Examples

```
add_pdn_stripe -grid main_grid -layer metal1 -followpins
add_pdn_stripe -grid main_grid -layer metal2 -width 0.17 -followpins
add pdn_stripe -grid main_grid -layer metal4 -width 0.48 -pitch 56.0 -offset 2 -starts_
↪with GROUND
```

Add rings

The add_pnd_ring command is used to define power/ground rings around a grid region. The ring structure is built using two layers that are orthogonal to each other. A power/ground pair will be added above and below the grid using the horizontal layer, with another power/ground pair to the left and right using the vertical layer. Together these 4 pairs of power/ground stripes form a ring around the specified grid. Power straps on these layers that are inside the enclosed region are extend to connect to the ring.

```
add_pdn_ring [-grid grid_name] \
             -layers layer_name \
             -widths (width_value|list_of_2_values) \
             -spacings (spacing_value|list_of_2_values) \
             [-core_offset offset_value] \
             [-pad_offset offset_value] \
             [-add_connect] \
             [-extend_to_boundary] \
             [-connect_to_pads] \
             [-connect_to_pad_layers layers] \
             [-starts_with (POWER|GROUND)] \
             [-nets list_of_nets]
```

Options

Switch Name	Description
-grid	Specifies the name of the grid to which this ring defintion will be added. (Default: Last grid created by defin_pdn_grid)
-layer	Specifies the name of the layer for these stripes
-width	Value for the width of the stdcell rail
-spacing	Optional specification of the spacing between power/ground pairs within a single pitch. (Default: pitch / 2)
-core_offset	Value for the offset of the ring from the grid region
-pad_offset	When defining a power grid for the top level of an SoC, can be used to define the offset of ring from the pad cells
-add_connect	Automatically add a connection between the two layers
-extend_to_boundary	Extend the rings to the grid boundary
-connect_to_pads	The core side of the pad pins will be connected to the ring
-connect_to_pad_layers	Restrict the pad pins layers to this list
-starts_with	Specifies whether the first strap placed will be POWER or GROUND (Default: grid setting)
-nets	Limit straps to just this list of nets

Examples

```
add_pdn_ring -grid main_grid -layer {metal6 metal7} -widths 5.0 -spacings  3.0 -core_
↪offset 5
```

Add connections

The add_pdn_connect command is used to define which layers in the power grid are to be connected together. During power grid generation, vias will be added for overlapping power nets and overlapping ground nets. The use of fixed vias from the technology file can be specified or else via stacks will be constructed using VIARULEs. If VIARULEs are not available in the technology, then fixed vias must be used.

```
add_pdn_connect [-grid grid_name] \
                [-layers list_of_two_layers] \
                [-cut_pitch pitch_value] \
                [-fixed_vias list_of_fixed_vias] \
                [-dont_use_vias list_of_vias] \
                [-max_rows rows] \
                [-max_columns columns] \
                [-ongrid ongrid_layers] \
                [-split_cuts split_cuts_mapping]
```

Options

Switch Name	Description
-grid	Specifies the name of the grid definition to which this connection will be added. (Default: Last grid created by define_pdn_grid)
-layers	Layers to be connected where there are overlapping power or overlapping ground nets
-cut_pitch	When the two layers are parallel e.g. overlapping stdcell rails, specify the distance between via cuts
-fixed_vias	List of fixed vias to be used to form the via stack
-dont_use_vias	List or pattern of vias to not use to form the via stack
-max_rows	Maximum number of rows when adding arrays of vias
-max_columns	Maximum number of columns when adding arrays of vias
-ongrid	List of intermediate layers in a via stack to snap onto a routing grid
-split_cuts	Specifies layers to use split cuts on with an associated pitch, for example {metal3 0.380 metal5 0.500}.

Examples

```
add_pdn_connect -grid main_grid -layers {metal1 metal2} -cut_pitch 0.16
add_pdn_connect -grid main_grid -layers {metal2 metal4}
add_pdn_connect -grid main_grid -layers {metal4 metal7}

add_pdn_connect -grid ram -layers {metal4 metal5}
add_pdn_connect -grid ram -layers {metal5 metal6}
add_pdn_connect -grid ram -layers {metal6 metal7}
```

(continues on next page)

(continued from previous page)

```
add_pdn_connect -grid rotated_rams -layers {metal4 metal6}
add_pdn_connect -grid rotated_rams -layers {metal6 metal7}
```

Build power grid

Build a power grid in accordance with the information specified.

```
pdngen [-skip_trim] \
       [-dont_add_pins] \
       [-reset] \
       [-ripup] \
       [-report_only] \
       [-failed_via_report file]
```

Options

Switch Name	Description
-skip_trim	Skip the metal trim step, which attempts to remove metal stubs
-dont_add_pins	Prevent the creation of block pins during
-reset	Reset the grid and domain specifications
-ripup	Ripup the existing power grid, as specified by the voltage domains
-report_only	Print the current specifications
-failed_via_report	Generate a report file which can be viewed in the DRC viewer for all the failed vias (ie. those that did not get built or were removed).

Repairing power grid vias after detailed routing

To remove vias which generate DRC violations after detailed placement and routing use `repair_pdn_vias`.

```
repair_pdn_vias [-all] \
                [-net net_name]
```

Options

Name	Description
-all	Repair vias on all supply nets
-net	Repair only vias on the specified net

Converting former PDNGEN configuration file to tcl commands

To get an initial conversion from the former PDNGEN configuration file to the current tcl commands use `convert_pdn_config`. This command will provide an initial set of commands based on the provided file, it is recommended that the user double-check the conversion to ensure nothing was missed.

```
convert_pdn_config config_file
```

Options

Name	Description
`config_file`	Path to the old configuration file

Example scripts

Defining a SoC power grid with pads

```
add_global_connection -net VDD -pin_pattern {^VDD$} -power
add_global_connection -net VDD -pin_pattern {^VDDPE$}
add_global_connection -net VDD -pin_pattern {^VDDCE$}
add_global_connection -net VSS -pin_pattern {^VSS$} -ground
add_global_connection -net VSS -pin_pattern {^VSSE$}

set_voltage_domain -power VDD -ground VSS

define_pdn_grid -name "Core"
add_pdn_ring -grid "Core" -layers {metal8 metal9} -widths 5.0 -spacings 2.0 -core_
↪offsets 4.5 -connect_to_pads

add_pdn_stripe -followpins -layer metal1 -extend_to_core_ring

add_pdn_stripe -layer metal4 -width 0.48 -pitch 56.0 -offset 2.0 -extend_to_core_ring
add_pdn_stripe -layer metal7 -width 1.40 -pitch 40.0 -offset 2.0 -extend_to_core_ring
add_pdn_stripe -layer metal8 -width 1.40 -pitch 40.0 -offset 2.0 -extend_to_core_ring
add_pdn_stripe -layer metal9 -width 1.40 -pitch 40.0 -offset 2.0 -extend_to_core_ring

add_pdn_connect -layers {metal1 metal4}
add_pdn_connect -layers {metal4 metal7}
add_pdn_connect -layers {metal7 metal8}
add_pdn_connect -layers {metal8 metal9}
add_pdn_connect -layers {metal9 metal10}

pdngen
```

Sroute

The `add_sroute_connect` command is employed for connecting pins located outside of a specific power domain to the power ring, especially in cases where multiple power domains are present. During `sroute`, multi-cut vias will be added for new connections. The use of fixed vias from the technology file should be specified for the connection using the `add_sroute_connect` command. The use of max_rows and max_columns defines the row and column limit for the via stack.

```
add_sroute_connect
    -layers list_of_2_layers
    -cut_pitch pitch_value
    [-net net]
    [-outerNet outerNet]
    [-fixed_vias list_of_vias]
    [-max_rows rows]
    [-max_columns columns]
    [-metalwidths metalwidths]
    [-metalspaces metalspaces]
    [-ongrid ongrid_layers]
    [-insts inst]
```

Options

Switch Name	Description
`-net`	The inner net where the power ring exists.
`-outerNet`	The outer net where instances/pins that need to get connected exist.
`-layers`	The metal layers for vertical stripes within inner power ring.
`-cut_pitch`	Distance between via cuts when the two layers are parallel, e.g., overlapping stdcell rails. (Default:200 200)
`-fixed_vias`	List of fixed vias to be used to form the via stack.
`-max_rows`	Maximum number of rows when adding arrays of vias. (Default:10)
`-max_columns`	Maximum number of columns when adding arrays of vias. (Default:10)
`-metalwidths`	Width for each metal layer.
`-metalspaces`	Spacing of each metal layer.
`-ongrid`	List of intermediate layers in a via stack to snap onto a routing grid.
`-insts`	List of all the instances that contain the pin that needs to get connected with power ring. (Default:nothing)

Examples

```
add_sroute_connect  -net "VIN" -outerNet "VDD" -layers {met1 met4} -cut_pitch {200 200} -
↪fixed_vias {M3M4_PR_M} -metalwidths {1000 1000} -metalspaces {800} -ongrid {met3 met4}␣
↪-insts "temp_analog_1.a_header_0"
```

Regression tests

Limitations

Currently the following assumptions are made:

1. The design is rectangular
2. The input floorplan includes the stdcell rows, placement of all macro blocks and IO pins.
3. The stdcells rows will be cut around macro placements

FAQs

License

Read UPF Utility

This module contains functionality to read, and modify information from Unified Power Format (UPF) files.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Read UPF

Sources the UPF file.

```
read_upf
    -file file
```

Options

Switch Name	Description
`-file`	Path to `.upf` file.

Create Power Domain

```
create_power_domain
    [-elements elements]
    name
```

Options

Switch Name	Description
-elements	List of module paths that belong this this domain OR * for top domain.
name	Domain name.

Create Logic Port

```
create_logic_port
    [-direction direction]
    port_name
```

Options

Switch Name	Description
-direction	Direction of the port (in, out, inout).
port_name	Port name.

Create Power Switch

```
create_power_switch
    [-domain domain]
    [-output_supply_port output_supply_port]
    [-input_supply_port input_supply_port]
    [-control_port control_port]
    [-on_state on_state]
    name
```

Options

Switch Name	Description
-domain	Power domain name.
-output_supply_port	Output supply port of the switch.
-input_supply_port	Input supply port of the switch.
-control_port	Control port on the switch.
-on_state	One of {state_name, input_supply_port, boolean_expression}.
name	Power switch name.

Create or Update Isolation Strategy

```
set_isolation
    [-domain domain]
    [-applies_to applies_to]
    [-clamp_value clamp_value]
    [-isolation_signal isolation_signal]
    [-isolation_sense isolation_sense]
    [-location location]
    [-update]
    name
```

Options

Switch Name	Description
-domain	Power domain
-applies_to	Restricts the strategy to apply one of these (inputs, outputs, both).
-clamp_value	Value the isolation can drive (0, 1).
-isolation_signal	The control signal for this strategy.
-isolation_sense	The active level of isolation control signal.
-location	Domain in which isolation cells are placed (parent, self, fanout).
-update	Only available if using existing strategy, will error if the strategy doesn't exist.
name	Isolation strategy name.

Set Interface cell

```
use_interface_cell
    [-domain domain]
    [-strategy strategy]
    [-lib_cells lib_cells]
```

Options

Switch Name	Description
-domain	Power domain name.
-strategy	Isolation strategy name.
-lib_cells	List of lib cells that could be used.

Set Domain Area

```
set_domain_area
    domain_name
    -area {llx lly urx ury}
```

Options

Switch Name	Description
domain_name	Power domain name.
-area	x-/y- coordinates in microns for the lower left and upper right corners of the power domain area.

Map existing power switch

```
map_power_switch
    [-switch_name_list switch_name_list]
    [-lib_cells lib_cells]
    [-port_map port_map]
```

Options

Switch Name	Description
-switch_name_list	A list of switches (as defined by create_power_switch) to map.
-lib_cells	A list of library cells that could be mapped to the power switch
-port_map	A map that associates model ports defined by create_power_switch to logical ports

Example scripts

Example script demonstrating how to run `upf` related commands can be found here:

```
./test/upf_test.tcl
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

License

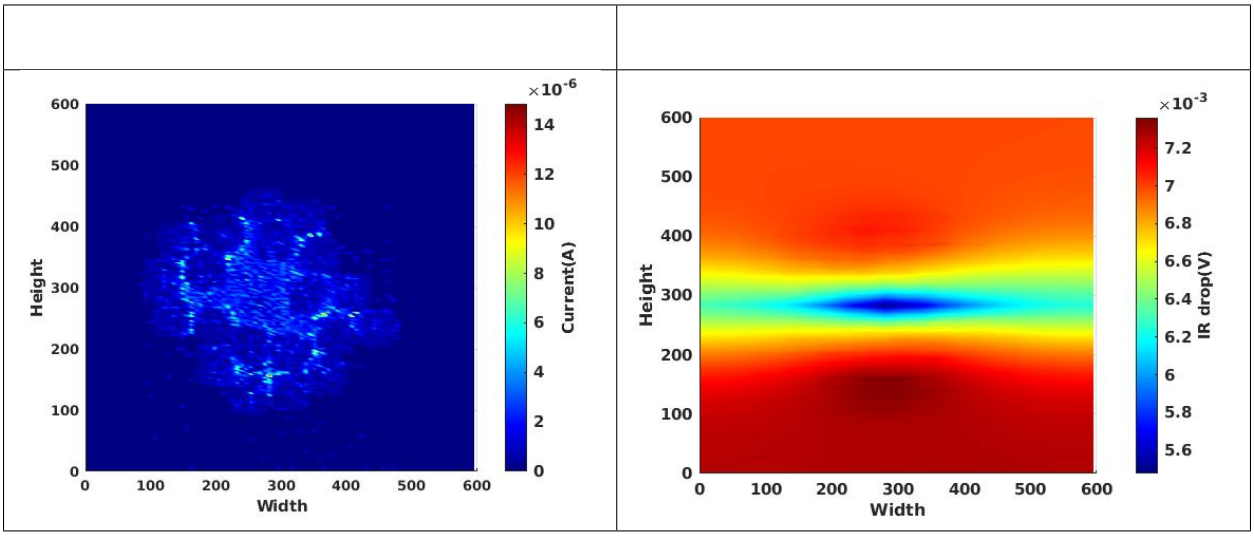
BSD 3-Clause License. See *LICENSE* file.

IR Drop Analysis

The IR Drop Analysis module in OpenROAD (`psm`) is based on PDNSim, an open-source static IR analyzer.

Features:

- Report worst IR drop.
- Report worst current density over all nodes and wire segments in the power distribution network, given a placed and PDN-synthesized design.
- Check for floating PDN stripes on the power and ground nets.
- Spice netlist writer for power distribution network wire segments.



Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Analyze Power Grid

```
analyze_power_grid
    [-vsrc vsrc_file]
    [-outfile out_file]
    [-error_file err_file]
    [-enable_em]
    [-em_outfile em_out_file]
    [-net net_name]
    [-dx bump_pitch_x]
    [-dy bump_pitch_y]
    [-node_density val_node_density]
    [-node_density_factor val_node_density_factor]
    [-corner corner]
```

Options

Switch Name	Description
`-vsrc`	File to set the location of the power C4 bumps/IO pins. Vsrc_aes.loc file for an example with a description specified *here*.
`-dx,-dy`	These arguments set the bump pitch to decide the voltage source location in the absence of a vsrc file. Default bump pitch of 140um used in absence of these arguments and vsrc.
`-net`	Name of the net to analyze, power or ground net name.
`-enable_em`	Report current per power grid segment.
`-outfile`	Write per-instance voltage into the file.
`-em_outfile`	Write the per-segment current values into a file. This option is only available if used in combination with `-enable_em`.
`-voltage`	Sets the voltage on a specific net. If this option is not set, the Liberty file's voltage value is obtained from operating conditions.
`-node_density`	Node density (in microns) on the standard cell rails. It cannot be used together with `-node_density_factor`.
`-node_density_factor`	Factor which is multiplied by standard cell height to determine the node density on the std cell rails. It cannot be used together with `-node_density`. The default value is `5`, and the allowed values are integers `[0, MAX_INT]`.
`-corner`	Corner to use for analysis.

Check Power Grid

```
check_power_grid -net net_name
```

Options

Switch Name	Description
-net	Name of the net to analyze. Must be a power or ground net name.

Write Spice Power Grid

```
write_pg_spice
    [-vsrc vsrc_file]
    [-outfile out_file]
    [-net net_name]
    [-dx bump_pitch_x]
    [-dy bump_pitch_y]
    [-corner corner]
```

Options

Switch Name	Description
-vsrc	File to set the location of the power C4 bumps/IO pins. See Vsrc_aes.loc file for an example and its *description*.
-dx,-dy	Set the bump pitch to decide the voltage source location in the absence of a vsrc file. The default bump pitch is 140um if neither these arguments nor a vsrc file are given.
-net	Name of the net to analyze. Must be a power or ground net name.
-outfile	Write per-instance voltage written into the file.
-corner	Corner to use for analysis.

Set PDNSim Net voltage

```
set_pdnsim_net_voltage
    [-net net_name]
    [-voltage volt]
```

Options

Switch Name	Description
`-net`	Name of the net to analyze. It must be a power or ground net name.
`-voltage`	Sets the voltage on a specific net. If this option is not given, the Liberty file's voltage value is obtained from operating conditions.

Useful Developer Commands

If you are a developer, you might find these useful. More details can be found in the source file or the swig file.

Command Name	Description
`find_net`	Get a reference to net name.

Example scripts

Example scripts demonstrating how to run PDNSim on a sample design on `aes` as follows:

```
./test/aes_test_vdd.tcl
./test/aes_test_vss.tcl
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

References

1. PDNSIM documentation
2. Chhabria, V.A. and Sapatnekar, S.S. (no date) The-openroad-project/pdnsim: Power Grid Analysis, GitHub. Available at: https://github.com/The-OpenROAD-Project/PDNSim (Accessed: 24 July 2023). (link)

License

BSD 3-Clause License. See LICENSE file.

Voltage source location file description

This file specifies the description of the C4 bump configurations file. The file is a csv as described below:

```
<x_coordinate>, <y_coordinate>, <octagonal_c4_bump_edge_length>, <voltage_value>
```

The x and y coordinate specify the center location of the voltage C4 bumps in micro meter.

The octagonal c4_edge_length specifies the edge length of the C4 to determine the pitch of the RDL layer in micron

Voltage_value specifies the value of voltage source at the C4 bump. In case there is a need to specify voltage drop in micron

Example file

```
250,250,20,1.1
130,170,20,1.1
370,410,10,1.1
410,450,10,1.1
```

5.2.17 RePlAce

RePlAce: Advancing Solution Quality and Routability Validation in Global Placement

Features:

- Analytic and nonlinear placement algorithm. Solves electrostatic force equations using Nesterov's method. (link)
- Verified with various commercial technologies and research enablements using OpenDB (7/14/16/28/45/55/65nm).
- Verified deterministic solution generation with various compilers and OS.
- Supports Mixed-size placement mode.

Visualized examples from ISPD 2006 contest; adaptec2.inf	Real-world Design: Coyote (TSMC16 7.5T)

Commands

```
global_placement
    [-timing_driven]
    [-routability_driven]
    [-skip_initial_place]
    [-incremental]
    [-bin_grid_count grid_count]
    [-density target_density]
```

(continues on next page)

(continued from previous page)

```
    [-init_density_penalty init_density_penalty]
    [-init_wirelength_coef init_wirelength_coef]
    [-min_phi_coef min_phi_conef]
    [-max_phi_coef max_phi_coef]
    [-overflow overflow]
    [-initial_place_max_iter initial_place_max_iter]
    [-initial_place_max_fanout initial_place_max_fanout]
    [-routability_check_overflow routability_check_overflow]
    [-routability_max_density routability_max_density]
    [-routability_max_bloat_iter routability_max_bloat_iter]
    [-routability_max_inflation_iter routability_max_inflation_iter]
    [-routability_target_rc_metric routability_target_rc_metric]
    [-routability_inflation_ratio_coef routability_inflation_ratio_coef]
    [-routability_pitch_scale routability_pitch_scale]
    [-routability_max_inflation_ratio routability_max_inflation_ratio]
    [-routability_rc_coefficients routability_rc_coefficients]
    [-timing_driven_net_reweight_overflow]
    [-timing_driven_net_weight_max]
    [-timing_driven_nets_percentage]
    [-pad_left pad_left]
    [-pad_right pad_right]
    [-verbose_level level]
    [-force_cpu]
```

```
cluster_flops
    [-tray_weight tray_weight]
    [-timing_weight timing_weight]
    [-max_split_size max_split_size]
```

Tuning Parameters

- `-timing_driven`: Enable timing-driven mode
- `-routability_driven`: Enable routability-driven mode
- `-skip_initial_place` : Skip the initial placement (BiCGSTAB solving) before Nesterov placement. IP improves HPWL by ~5% on large designs. Equal to '-initial_place_max_iter 0'
- `-incremental` : Enable the incremental global placement. Users would need to tune other parameters (e.g., init_density_penalty) with pre-placed solutions.
- `-bin_grid_count`: set bin grid's counts. Default value is defined by internal heuristic. Allowed values are `[64,128,256,512,..., int]`.
- `-density`: set target density. Default value is 0.70. Allowed values are `[0-1, float]`.
- `-init_density_penalty`: set initial density penalty. Default value is 8e-5. Allowed values are `[1e-6 - 1e6, float]`.
- `-init_wirelength_coef`: set initial wirelength coefficient. Default value is 0.25. Allowed values are `[unlimited, float]`.
- `-min_phi_coef`: set `pcof_min(μ_k Lower Bound)`. Default value is 0.95. Allowed values are `[0.95-1.05, float]`.

- -max_phi_coef: set pcof_max(μ_k Upper Bound). Default value is 1.05. Allowed values are [1.00-1.20, float].
- -overflow: set target overflow for termination condition. Default value is 0.1. Allowed values are [0-1, float].
- -initial_place_max_iter: set maximum iterations in initial place. Default value is 20. Allowed values are [0-MAX_INT, int].
- -initial_place_max_fanout: set net escape condition in initial place when 'fanout >= initial_place_max_fanout'. Default value is 200. Allowed values are [1-MAX_INT, int].
- -timing_driven_net_reweight_overflow: set overflow threshold for timing-driven net reweighting. Allowed values are tcl list of [0-100, int].
- -timing_driven_net_weight_max: Set the multiplier for the most timing critical nets. Default value is 1.9.
- -timing_driven_nets_percentage: Set the percentage of nets that are reweighted in timing-driven mode. Default value is 10. Allowed values are [0-100, float]
- -verbose_level: set verbose level for RePlAce. Default value is 1. Allowed values are [0-5, int].
- -force_cpu: Force to use the CPU solver even if the GPU is available.
- tray_weight: Set the weighting factor for tray cost in flip-flop clustering (recommended to be [20.0, float]).
- timing_weight: Set the weighting factor for timing-critical paths in flip-flop clusering (recommended to be [1.0. float]).
- max_split_size: The maximum size of a single pointset after running the pointset decomposition algorithm for runtime improvement in flop clustering (to not run pointset decomposition, set as -1).

-timing_driven does a virtual repair_design to find slacks and weight nets with low slack. It adjusts the worst slacks (10% by default, modified with -timing_driven_nets_percentage) using a multiplier (1.9 by default, modified with -timing_driven_net_weight_max). The multiplier is scaled from the full value for the worst slack, to 1.0 at the timing_driven_nets_percentage point. Use the set_wire_rc command to set resistance and capacitance of estimated wires used for timing.

Example scripts

Regression tests

Limitations

Using the Python interface to gpl

This api tries to stay close to the api defined in C++ class Replace that is located in gpl/include/gpl/Replace.h

When initializing a design, a sequence of Python commands might look like the following:

```
from openroad import Design, Tech
tech = Tech()
tech.readLef(...)
design = Design(tech)
design.readDef(...)
gpl = design.getReplace()
```

Here is an example of some options / configurations to the global placer. (See Replace.h for a complete list)

```
gpl.setInitialPlaceMaxIter(iter)
gpl.setSkipIoMode(skip_io)
gpl.setTimingDrivenMode(timing_driven)
gpl.setTimingNetWeightMax(weight)
```

There are some useful Python functions located in the file grt/test/grt_aux.py but these are not considered a part of the (final) api and they may change.

FAQs

Check out GitHub discussion about this tool.

External references

- C.-K. Cheng, A. B. Kahng, I. Kang and L. Wang, "RePlAce: Advancing Solution Quality and Routability Validation in Global Placement", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 38(9) (2019), pp. 1717-1730.
- J. Lu, P. Chen, C.-C. Chang, L. Sha, D. J.-H. Huang, C.-C. Teng and C.-K. Cheng, "ePlace: Electrostatics based Placement using Fast Fourier Transform and Nesterov's Method", ACM TODAES 20(2) (2015), article 17.
- J. Lu, H. Zhuang, P. Chen, H. Chang, C.-C. Chang, Y.-C. Wong, L. Sha, D. J.-H. Huang, Y. Luo, C.-C. Teng and C.-K. Cheng, "ePlace-MS: Electrostatics based Placement for Mixed-Size Circuits", IEEE TCAD 34(5) (2015), pp. 685-698.
- A. B. Kahng, J. Li and L. Wang,
  "Improved Flop Tray-Based Design Implementation for Power Reduction",
  IEEE/ACM ICCAD, 2016, pp. 20:1-20:8.
- The timing-driven mode has been implemented by Mingyu Woo (only available in legacy repo in standalone branch.)
- The routability-driven mode has been implemented by Mingyu Woo.
- Timing-driven mode re-implementation is ongoing with the current clean-code structure.

Authors

- Authors/maintainer since Jan 2020: Mingyu Woo (Ph.D. Advisor: Andrew. B. Kahng)
- Original open-sourcing of RePlAce: August 2018, by Ilgweon Kang (Ph.D. Advisor: Chung-Kuan Cheng), Lutong Wang (Ph.D. Advisor: Andrew B. Kahng), and Mingyu Woo (Ph.D. Advisor: Andrew B. Kahng).
- Also thanks to Dr. Jingwei Lu for open-sourcing the previous ePlace-MS/ePlace project code.

License

BSD 3-Clause License. See LICENSE file.

OpenROAD Tcl Usage (global_placement)

```
global_placement
    [-skip_initial_place]
    [-incremental]
    [-bin_grid_count grid_count]
    [-density density]
    [-init_density_penalty init_density_penalty]
    [-init_wirelength_coef init_wirelength_coef]
    [-min_phi_coef min_phi_coef]
    [-max_phi_coef max_phi_coef]
    [-overflow overflow]
    [-initial_place_max_iter max_iter]
    [-initial_place_max_fanout max_fanout]
    [-verbose_level verbose_level]
```

Flow Control

- **skip_initial_place** : Skip the initial placement (BiCGSTAB solving) before Nesterov placement. IP improves HPWL by ~5% on large designs.
- **incremental** : Enable the incremental global placement. Users would need to tune other parameters (e.g. init_density_penalty) with the pre-placed solutions.

Tuning Parameters

- **bin_grid_count** : Set bin grid's count manually. Default: Defined by internal algorithm. [64,128,256,512,..., int]
- **density** : Set target density. Default: 0.70 [0-1, float]
- **init_density_penalty** : Set initial density penalty. Default : 8e-5 [1e-6 - 1e6, float]
- **min_phi_coef** : Set pcof_min(μ_k Lower Bound). Default: 0.95 [0.95-1.05, float]
- **max_phi_coef** : Set pcof_max(μ_k Upper Bound). Default: 1.05 [1.00-1.20, float]
- **overflow** : Set target overflow for termination condition. Default: 0.1 [0-1, float]

Other Options

- **verbose_level** [0-10, int] : Set verbose level for RePlAce. Default: 1

Note that all of the TCL commands are defined in the .../src/replace.tcl and .../src/replace.i.

5.2.18 Gate Resizer

Gate Resizer commands are described below. The `resizer` commands stop when the design area is `-max_utilization util` percent of the core area. `util` is between 0 and 100. The `resizer` stops and reports an error if the max utilization is exceeded.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Set Wire RC

The `set_wire_rc` command sets the resistance and capacitance used to estimate delay of routing wires. Separate values can be specified for clock and data nets with the `-signal` and `-clock` flags. Without either `-signal` or `-clock` the resistance and capacitance for clocks and data nets are set.

```
set_wire_rc
    [-clock]
    [-signal]
    [-layer layer_name]

or
set_wire_rc
    [-resistance res]
    [-capacitance cap]
```

Options

Switch Name	Description
`-clock`	Enable setting of RC for clock nets.
`-signal`	Enable setting of RC for signal nets.
`-layer`	Use the LEF technology resistance and area/edge capacitance values for the layer. This is used for a default width wire on the layer.
`-resistance`	Resistance per unit length, units are from the first Liberty file read, usually in the form of $\frac{resistanceUnit}{distanceUnit}$. Usually k/µm.
`-capacitance`	Capacitance per unit length, units are from the first Liberty file read, usually in the form of $\frac{capacitanceUnit}{distanceUnit}$. Usually pF/µm.

Set Layer RC

The `set_layer_rc` command can be used to set the resistance and capacitance for a layer or via. This is useful if these values are missing from the LEF file, or to override the values in the LEF.

```
set_layer_rc
    [-layer layer]
    [-via via_layer]
    [-resistance res]
    [-capacitance cap]
    [-corner corner]
```

Options

Switch Name	Description
`-layer`	Set layer name to modify. Note that the layer must be a routing layer.
`-via`	Select via layer name. Note that via resistance is per cut/via, not area-based.
`-resistance`	Resistance per unit length, same convention as `set_wire_rc`.
`-capacitance`	Capacitance per unit length, same convention as `set_wire_rc`.
`-corner`	Process corner to use.

Estimate Parasitics

Estimate RC parasitics based on placed component pin locations. If there are no component locations, then no parasitics are added. The resistance and capacitance values are per distance unit of a routing wire. Use the `set_units` command to check units or `set_cmd_units` to change units. The goal is to represent "average" routing layer resistance and capacitance. If the set_wire_rc command is not called before resizing, then the default_wireload model specified in the first Liberty file read or with the SDC set_wire_load command is used to make parasitics.

After the `global_route` command has been called, the global routing topology and layers can be used to estimate parasitics with the `-global_routing` flag.

```
estimate_parasitics
    -placement|-global_routing
```

Options

Switch Name	Description
`-placement` or `-global_routing`	Either of these flags must be set. Parasitics are estimated based after placement stage versus after global routing stage.

Set Don't Use

The `set_dont_use` command removes library cells from consideration by the `resizer`. `lib_cells` is a list of cells returned by `get_lib_cells` or a list of cell names (`wildcards` allowed). For example, `DLY*` says do not use cells with names that begin with `DLY` in all libraries.

```
set_dont_use lib_cells
unset_dont_use lib_cells
```

Set Don't Touch

The `set_dont_touch` command prevents the resizer commands from modifying instances or nets.

```
set_dont_touch instances_nets
unset_dont_touch instances_nets
```

Buffer Ports

The `buffer_ports -inputs` command adds a buffer between the input and its loads. The `buffer_ports -outputs` adds a buffer between the port driver and the output port. Inserting buffers on input and output ports makes the block input capacitances and output drives independent of the block internals.

```
buffer_ports
    [-inputs]
    [-outputs]
    [-max_utilization util]
```

Options

Switch Name	Description
`-inputs`, `-outputs`	Insert a buffer between the input and load, output and load respectively. The default behavior is `-inputs` and `-outputs` set if neither is specified.
`-max_utilization`	Defines the percentage of core area used.

Remove Buffers

Use the `remove_buffers` command to remove buffers inserted by synthesis. This step is recommended before using `repair_design` so that there is more flexibility in buffering nets.

```
remove_buffers
```

Repair Design

The `repair_design` command inserts buffers on nets to repair max slew, max capacitance and max fanout violations, and on long wires to reduce RC delay in the wire. It also resizes gates to normalize slews. Use `estimate_parasitics -placement` before `repair_design` to estimate parasitics considered during repair. Placement-based parasitics cannot accurately predict routed parasitics, so a margin can be used to "over-repair" the design to compensate.

```
repair_design
    [-max_wire_length max_length]
    [-slew_margin slew_margin]
    [-cap_margin cap_margin]
    [-max_utilization util]
    [-verbose]
```

Options

Switch Name	Description
`-max_wire_length`	Maximum length of wires (in microns), defaults to a value that minimizes the wire delay for the wire RC values specified by `set_wire_rc`.
`-slew_margin`	Add a slew margin. The default value is `0`, the allowed values are integers `[0, 100]`.
`-cap_margin`	Add a capactitance margin. The default value is `0`, the allowed values are integers `[0, 100]`.
`-max_utilization`	Defines the percentage of core area used.
`-verbose`	Enable verbose logging on progress of the repair.

Repair Tie Fanout

The `repair_tie_fanout` command connects each tie high/low load to a copy of the tie high/low cell.

```
repair_tie_fanout
    [-separation dist]
    [-verbose]
    lib_port
```

Options

Switch Name	Description
`-separation`	Tie high/low insts are separated from the load by this value (Liberty units, usually microns).
`-verbose`	Enable verbose logging of repair progress.
`lib_port`	Tie high/low port, which can be a library/cell/port name or object returned by `get_lib_pins`.

Repair Timing

The `repair_timing` command repairs setup and hold violations. It should be run after clock tree synthesis with propagated clocks. Setup repair is done before hold repair so that hold repair does not cause setup checks to fail.

The worst setup path is always repaired. Next, violating paths to endpoints are repaired to reduced the total negative slack.

```
repair_timing
    [-setup]
    [-hold]
    [-setup_margin setup_margin]
    [-hold_margin hold_margin]
    [-allow_setup_violations]
    [-repair_tns tns_end_percent]
    [-max_utilization util]
    [-max_buffer_percent buffer_percent]
    [-verbose]
```

Options

Switch Name	Description
`-setup`	Repair setup timing.
`-hold`	Repair hold timing.
`-setup_margin`	Add additional setup slack margin.
`-hold_margin`	Add additional hold slack margin.
`-allow_setup_violations`	While repairing hold violations, buffers are not inserted that will cause setup violations unless `-allow_setup_violations` is specified.
`-repair_tns`	Percentage of violating endpoints to repair (0-100). When `tns_end_percent` is zero (the default), only the worst endpoint is repaired. When `tns_end_percent` is 100, all violating endpoints are repaired.
`-max_utilization`	Defines the percentage of core area used.
`-max_buffer_percent`	Specify a maximum number of buffers to insert to repair hold violations as a percentage of the number of instances in the design. The default value is `20`, and the allowed values are integers `[0, 100]`.
`-verbose`	Enable verbose logging of the repair progress.

Use `-recover_power` to specify the percent of paths with positive slack which will be considered for gate resizing to save power. It is recommended that this option be used with global routing based parasitics.

Repair Clock Nets

The `clock_tree_synthesis` command inserts a clock tree in the design but may leave a long wire from the clock input pin to the clock tree root buffer. The `repair_clock_nets` command inserts buffers in the wire from the clock input pin to the clock root buffer.

```
repair_clock_nets
    [-max_wire_length max_wire_length]
```

Options

Switch Name	Description
-max_wire_length	Maximum length of wires (in microns), defaults to a value that minimizes the wire delay for the wire RC values specified by set_wire_rc.

Repair Clock Inverters

```
repair_clock_inverters
```

Report Design Area

The report_design_area command reports the area of the design's components and the utilization.

```
report_design_area
```

Report Floating Nets

The report_floating_nets command reports nets with only one pin connection.

```
report_floating_nets
    [-verbose]
```

Options

Switch Name	Description
-verbose	Print the net names.

Useful Developer Commands

If you are a developer, you might find these useful. More details can be found in the source file or the swig file.

Command Name	Description
repair_setup_pin	Repair setup pin violation.
check_parasitics	Check if the estimate_parasitics command has been called.
parse_time_margin_arg	Get the raw value for timing margin (e.g. slack_margin, setup_margin, hold_margin)
parse_percent_margin_arg	Get the above margin in perentage format.
parse_margin_arg	Same as parse_percent_margin_arg.
parse_max_util	Check maximum utilization.
parse_max_wire_length	Get maximum wirelength.
check_corner_wire_caps	Check wire capacitance for corner.
check_max_wire_length	Check if wirelength is allowed by rsz for minimum delay.
dblayer_wire_rc	Get layer RC values.
set_dblayer_wire_rc	Set layer RC values.

Example scripts

A typical `resizer` command file (after a design and Liberty libraries have been read) is shown below.

```
read_sdc gcd.sdc

set_wire_rc -layer metal2

set_dont_use {CLKBUF_* AOI211_X1 OAI211_X1}

buffer_ports
repair_design -max_wire_length 100
repair_tie_fanout LOGIC0_X1/Z
repair_tie_fanout LOGIC1_X1/Z
# clock tree synthesis...
repair_timing
```

Note that OpenSTA commands can be used to report timing metrics before or after resizing the design.

```
set_wire_rc -layer metal2
report_checks
report_tns
report_wns
report_checks

repair_design

report_checks
report_tns
report_wns
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

License

BSD 3-Clause License. See *LICENSE* file.

5.2.19 Detailed Placement

The detailed placement module in OpenROAD (`dpl`) is based on OpenDP, or Open-Source Detailed Placement Engine. Its key features are:

- Fence region.
- Fragmented ROWs.

Commands

Detailed Placement

The `detailed_placement` command performs detailed placement of instances to legal locations after global placement.

```
detailed_placement
    [-max_displacement disp|{disp_x disp_y}]
    [-disallow_one_site_gaps]
    [-report_file_name filename]
```

Options

Switch Name	Description
`-max_displacement`	Maximum distance that an instance can be moved (in microns) when finding a site where it can be placed. Either set one value for both directions or set `{disp_x disp_y}` for individual directions. The default values are `{500, 100}`, and the allowed values within are integers `[0, MAX_INT]`.
`-disallow_one_site_gaps`	Disallow one site gap during placement check.
`-report_file_name`	Filename for saving the report to (e.g. `report.json`.

Set Placement Padding

The `set_placement_padding` command sets left and right padding in multiples of the row site width. Use the `set_placement_padding` command before legalizing placement to leave room for routing. Use the `-global` flag for padding that applies to all instances. Use `-instances` for instance-specific padding. The instances `insts` can be a list of instance names, or an instance object returned by the SDC `get_cells` command. To specify padding for all instances of a common master, use the `-filter` "ref_name == " option to `get_cells`.

```
set_placement_padding
    -global|-masters masters|-instances insts
    [-right site_count]
    [-left site_count]
```

Options

Warning: Either one of these flags must be set: `-global | -masters | -instances`. The order of preference is `global > masters > instances`

Switch Name	Description
`-global`	Set padding globally using `left` and `right` values.
`-masters`	Set padding only for these masters using `left` and `right` values.
`-instances`	For `-instances`, you will set padding only for these insts using `left` and `right` values.
`-left`	Left padding (in site count).
`-right`	Right padding (in site count).
`instances`	Set padding for these list of instances. Not to be confused with the `-instances` switch above.

Filler Placement

The `filler_placement` command fills gaps between detail-placed instances to connect the power and ground rails in the rows. `filler_masters` is a list of master/macro names to use for filling the gaps. Wildcard matching is supported, so `FILL*` will match, e.g., `FILLCELL_X1 FILLCELL_X16 FILLCELL_X2 FILLCELL_X32 FILLCELL_X4 FILLCELL_X8`. To specify a different naming prefix from `FILLER_` use `-prefix <new prefix>`.

```
filler_placement
    [-prefix prefix]
    filler_masters
```

Options

Switch Name	Description
`-prefix`	Prefix to name the filler cells. The default value is `FILLER_`.
`filler_masters`	Filler master cells.

Remove Fillers

This command removes all filler cells.

```
remove_fillers
```

No arguments are needed for this function.

Check Placement

The `check_placement` command checks the placement legality. It returns `0` if the placement is legal.

```
check_placement
    [-verbose]
    [-disallow_one_site_gaps]
    [-report_filename filename]
```

Options

Switch Name	Description
`-verbose`	Enable verbose logging.
`-disallow_one_site_gaps`	Disable one site gap during placement check.
`-report_file_name`	File name for saving the report to (e.g. `report.json`.

Optimize Mirroring

The `optimize_mirroring` command mirrors instances about the Y axis in a weak attempt to reduce the total half-perimeter wirelength (HPWL).

```
optimize_mirroring
```

No arguments are needed for this function.

Useful Developer Commands

If you are a developer, you might find these useful. More details can be found in the source file or the swig file.

Command Name	Description
`detailed_placement_debug`	Debug detailed placement.
`get_masters_arg`	Get masters from a design.
`get_inst_bbox`	Get bounding box of an instance.
`get_inst_grid_bbox`	Get grid bounding box of an instance.
`format_grid`	Format grid (takes in length `x` and site width `w` as inputs).
`get_row_site`	Get row site name.

Example scripts

Examples scripts demonstrating how to run `dpl` on a sample design of `aes` as follows:

```
./test/aes.tcl
```

Regression tests

There are a set of regression tests in `./test`. Refer to this *section* for more information.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

Authors

- SangGi Do and Mingyu Woo (respective Ph. D. advisors: Seokhyeong Kang, Andrew B. Kahng).
- Rewrite and port to OpenDB/OpenROAD by James Cherry, Parallax Software

References

1. Do, S., Woo, M., & Kang, S. (2019, May). Fence-region-aware mixed-height standard cell legalization. In Proceedings of the 2019 on Great Lakes Symposium on VLSI (pp. 259-262). (.pdf)

License

5.2.20 Clock Tree Synthesis

The clock tree synthesis module in OpenROAD (`cts`) is based on TritonCTS 2.0. It is available from the `clock_tree_synthesis` command. TritonCTS 2.0 performs on-the-fly characterization. Thus, there is no need to generate characterization data. The on-the-fly characterization feature can be optionally controlled by parameters specified by the `configure_cts_characterization` command. Use `set_wire_rc` command to set the clock routing layer.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Configure CTS Characterization

```
configure_cts_characterization
    [-max_slew max_slew]
    [-max_cap max_cap]
    [-slew_steps slew_steps]
    [-cap_steps cap_steps]
```

Options

Switch Name	Description
-max_slew	Max slew value (in the current time unit) that the characterization will test. If this parameter is omitted, the code would use max slew value for specified buffer in `buf_list` from liberty file.
-max_cap	Max capacitance value (in the current capacitance unit) that the characterization will test. If this parameter is omitted, the code would use max cap value for specified buffer in `buf_list` from liberty file.
-slew_steps	Number of steps that `max_slew` will be divided into for characterization. The default value is `12`, and the allowed values are integers `[0, MAX_INT]`.
-cap_steps	Number of steps that `max_cap` will be divided into for characterization. The default value is `34`, and the allowed values are integers `[0, MAX_INT]`.

Clock Tree Synthesis

```
clock_tree_synthesis
    -buf_list <list_of_buffers>
    [-root_buf root_buf]
    [-wire_unit wire_unit]
    [-clk_nets <list_of_clk_nets>]
    [-distance_between_buffers]
    [-branching_point_buffers_distance]
    [-clustering_exponent]
    [-clustering_unbalance_ratio]
    [-sink_clustering_enable]
    [-sink_clustering_size cluster_size]
    [-sink_clustering_max_diameter max_diameter]
    [-balance_levels]
    [-num_static_layers]
    [-sink_clustering_buffer]
```

Options

Switch Name	Description
-buf_list	Tcl list of master cells (buffers) that will be considered when making the wire segments (e.g. {BUFXX, BUFYY}).
-root_buffer	The master cell of the buffer that serves as root for the clock tree. If this parameter is omitted, the first master cell from -buf_list is taken.
-wire_unit	Minimum unit distance between buffers for a specific wire. If this parameter is omitted, the code gets the value from ten times the height of -root_buffer.
-clk_nets	String containing the names of the clock roots. If this parameter is omitted, cts automatically looks for the clock roots automatically.
-distance_between_buffers	Distance (in microns) between buffers that cts should use when creating the tree. When using this parameter, the clock tree algorithm is simplified and only uses a fraction of the segments from the LUT.
-branching_point_buffers_distance	Distance (in microns) that a branch has to have in order for a buffer to be inserted on a branch end-point. This requires the -distance_between_buffers value to be set.
-clustering_exponent	Value that determines the power used on the difference between sink and means on the CKMeans clustering algorithm. The default value is 4, and the allowed values are integers [0, MAX_INT].
-clustering_unbalance_ratio	Value determining each cluster's maximum capacity during CKMeans. A value of 0.5 (i.e., 50%) means that each cluster will have exactly half of all sinks for a specific region (half for each branch). The default value is 0.6, and the allowed values are floats [0, 1.0].
-sink_clustering_enable	Enable sink-clustering of sinks to create one level of sub-tree before building H-tree. Each cluster is driven by buffer which becomes end point of H-tree structure.
-sink_clustering_size	Specifies the maximum number of sinks per cluster. The default value is 20, and the allowed values are integers [0, MAX_INT].
-sink_clustering_max_diameter	Specifies maximum diameter (in microns) of sink cluster. The default value is 50, and the allowed values are integers [0, MAX_INT].
-balance_levels	Attempt to keep a similar number of levels in the clock tree across non-register cells (e.g., clock-gate or inverter). The default value is False, and the allowed values are bool.
-clk_nets	String containing the names of the clock roots. If this parameter is omitted, cts looks for the clock roots automatically.
-num_static_layers	Sets the number of static layers. The default value is 0, and the allowed values are integers [0, MAX_INT].
-sink_clustering_buffer	Set the sink clustering buffer(s) to be used.
-obstruction_aware	Enables obstruction-aware buffering such that clock buffers are not placed on top of blockages or hard macros. This option may reduce legalizer displacement, leading to better latency, skew or timing QoR. The default value is False, and the allowed values are bool.
-apply_ndr	Applies 2X spacing non-default rule to all clock nets except leaf-level nets. The default value is False.
-insertion_delay	Considers insertion delays in macro timing models to improve clustering. The default value is False.

Report CTS

Another command available from `cts` is `report_cts`. It is used to extract metrics after a successful `clock_tree_synthesis` run. These are:

- Number of Clock Roots
- Number of Buffers Inserted
- Number of Clock Subnets
- Number of Sinks.

```
report_cts
    [-out_file file]
```

Options

Switch Name	Description
`-out_file`	The file to save `cts` reports. If this parameter is omitted, the report is streamed to `stdout` and not saved.

Useful Developer Commands

If you are a developer, you might find these useful. More details can be found in the source file or the swig file.

Command Name	Description
`clock_tree_synthesis_debug`	Option to plot the CTS to GUI.

Example scripts

```
clock_tree_synthesis -root_buf "BUF_X4" \
                     -buf_list "BUF_X4" \
                     -wire_unit 20
report_cts "file.txt"
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

References

1. LEMON - **L**ibrary for **E**fficient **M**odeling and **O**ptimization in **N**etworks
2. Kahng, A. B., Li, J., & Wang, L. (2016, November). Improved flop tray-based design implementation for power reduction. In 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 1-8). IEEE. (.pdf)

Authors

TritonCTS 2.0 is written by Mateus Fogaça, PhD student in the Graduate Program on Microelectronics from the Federal University of Rio Grande do Sul (UFRGS), Brazil. Mr. Fogaça's advisor is Prof. Ricardo Reis.

Many guidance provided by (alphabetic order):

- Andrew B. Kahng
- Jiajia Li
- Kwangsoo Han
- Tom Spyrou

License

5.2.21 Global Routing

The global routing module in OpenROAD (`grt`) is based on FastRoute, an open-source global router originally derived from Iowa State University's FastRoute4.1 algorithm.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Global Route

```
global_route
    [-guide_file out_file]
    [-congestion_iterations iterations]
    [-congestion_report_file file_name]
    [-congestion_report_iter_step steps]
    [-grid_origin {x y}]
    [-critical_nets_percentage percent]
    [-allow_congestion]
    [-verbose]
    [-start_incremental]
    [-end_incremental]
```

Options

Switch Name	Description
`-guide_file`	Set the output guides file name (e.g., `route.guide`).
`-congestion_iterations`	Set the number of iterations made to remove the overflow of the routing. The default value is `50`, and the allowed values are integers `[0, MAX_INT]`.
`-congestion_report_file`	Set the file name to save the congestion report. The file generated can be read by the DRC viewer in the GUI (e.g., `report_file.rpt`).
`-congestion_report_iter_step`	Set the number of iterations to report. The default value is `0`, and the allowed values are integers `[0, MAX_INT]`.
`-grid_origin`	Set the (x, y) origin of the routing grid in DBU. For example, `-grid_origin {1 1}` corresponds to the die (0, 0) + 1 DBU in each x–, y- direction.
`-critical_nets_percentage`	Set the percentage of nets with the worst slack value that are considered timing critical, having preference over other nets during congestion iterations (e.g. `-critical_nets_percentage 30`). The default value is `0`, and the allowed values are integers `[0, MAX_INT]`.
`-allow_congestion`	Allow global routing results to be generated with remaining congestion. The default is false.
`-verbose`	This flag enables the full reporting of the global routing.
`-start_incremental`	This flag initializes the GRT listener to get the net modified. The default is false.
`-end_incremental`	This flag run incremental GRT with the nets modified. The default is false.

Set Routing Layers

```
set_routing_layers
    [-signal min-max]
    [-clock min-max]
```

Options

Switch Name	Description
`-signal`	Set the min and max routing signal layer (names) in this format "%s-%s".
`-clock`	Set the min and max routing clock layer (names) in this format "%s-%s".

Example: `set_routing_layers -signal Metal2-Metal10 -clock Metal6-Metal9`

Set Macro Extension

```
set_macro_extension extension
```

Options

Argument Name	Description
`extension`	Number of `GCells` added to the blockage boundaries from macros. A `GCell` is typically defined in terms of `Mx` routing tracks. The default `GCell` size is 15 `M3` pitches.

Example: `set_macro_extension 2`

Set Pin Offset

```
set_pin_offset offset
```

Options

Argument Name	Description
`offset`	Pin offset in microns (must be a positive integer).

Set Global Routing Layer Adjustment

The `set_global_routing_layer_adjustment` command sets routing resource adjustments in the routing layers of the design. Such adjustments reduce the number of routing tracks that the global router assumes to exist. This promotes the spreading of routing and reduces peak congestion, to reduce challenges for detailed routing.

```
set_global_routing_layer_adjustment layer adjustment
```

Options

Argument Name	Description
`layer`	Integer for the layer number (e.g. for M1 you would use 1).
`adjustment`	Float indicating the percentage reduction of each edge in the specified layer.

You can set adjustment for a specific layer, e.g., `set_global_routing_layer_adjustment Metal4 0.5` reduces the routing resources of routing layer `Metal4` by 50%. You can also set adjustment for all layers at once using `*`, e.g., `set_global_routing_layer_adjustment * 0.3` reduces the routing resources of all routing layers by 30%. And, you can also set resource adjustment for a layer range, e.g.: `set_global_routing_layer_adjustment Metal4-Metal8 0.3` reduces the routing resources of routing layers `Metal4`, `Metal5`, `Metal6`, `Metal7` and `Metal8` by 30%.

Set Routing Alpha

By default the global router uses heuristic rectilinear Steiner minimum trees (RSMTs) as an initial basis to construct route guides. An RSMT tries to minimize the total wirelength needed to connect a given set of pins. The Prim-Dijkstra heuristic is an alternative net topology algorithm that supports a trade-off between total wirelength and maximum path depth from the net driver to its loads. The `set_routing_alpha` command enables the Prim/Dijkstra algorithm and sets the alpha parameter used to trade-off wirelength and path depth. Alpha is between 0.0 and 1.0. When alpha is 0.0 the net topology minimizes total wirelength (i.e. capacitance). When alpha is 1.0 it minimizes longest path between the driver and loads (i.e., maximum resistance). Typical values are 0.4-0.8. You can call it multiple times for different nets.

```
set_routing_alpha
    [-net net_name]
    [-min_fanout fanout]
    [-min_hpwl hpwl]
    alpha
```

Options

Switch Name	Description
`-net`	Net name.
`-min_fanout`	Set the minimum number for fanout.
`-min_hpwl`	Set the minimum half-perimetere wirelength (microns).
`alpha`	Float between 0 and 1 describing the trade-off between wirelength and path depth.

Example: `set_routing_alpha -net clk 0.3` sets the alpha value of 0.3 for net *clk*.

Set Global Routing Region Adjustment

```
set_global_routing_region_adjustment
    {lower_left_x lower_left_y upper_right_x upper_right_y}
    -layer layer
    -adjustment adjustment
```

Options

Switch Name	Description
lower_left_x, lower_left_y, upper_right_x , upper_right_y	Bounding box to consider.
-layer	Integer for the layer number (e.g. for M1 you would use 1).
-adjustment	Float indicating the percentage reduction of each edge in the specified layer.

Example: `set_global_routing_region_adjustment {1.5 2 20 30.5} -layer Metal4 -adjustment 0.7`

Set Global Routing Randomness

The randomized global routing shuffles the order of the nets and randomly subtracts or adds to the capacities of a random set of edges.

```
set_global_routing_random
    [-seed seed]
    [-capacities_perturbation_percentage percent]
    [-perturbation_amount value]
```

Options

Switch Name	Description
-seed	Sets the random seed (must be non-zero for randomization).
-capacities_perturbation_percentage	Sets the percentage of edges whose capacities are perturbed. By default, the edge capacities are perturbed by adding or subtracting 1 (track) from the original capacity.
-perturbation_amount	Sets the perturbation value of the edge capacities. This option is only meaningful when -capacities_perturbation_percentage is used.

Example: `set_global_routing_random -seed 42 \ -capacities_perturbation_percentage 50 \ -perturbation_amount 2`

Set Specific Nets to Route

The `set_nets_to_route` command defines a list of nets to route. Only the nets defined in this command are routed, leaving the remaining nets without any global route guides.

```
set_nets_to_route
    net_names
```

Options

Switch Name	Description
`net_names`	Tcl list of set of nets (e.g. `{net1, net2}`).

Repair Antennas

The `repair_antennas` command checks the global routing for antenna violations and repairs the violations by inserting diodes near the gates of the violating nets. By default the command runs only one iteration to repair antennas. Filler instances added by the `filler_placement` command should NOT be in the database when `repair_antennas` is called.

```
repair_antennas
    [diode_cell]
    [-iterations iterations]
    [-ratio_margin margin]
```

Options

Switch Name	Description
`diode_cell`	Diode cell to fix antenna violations.
`-iterations`	Number of iterations. The default value is `1`, and the allowed values are integers `[0, MAX_INT]`.
`-ratio_margin`	Add a margin to the antenna ratios. The default value is `0`, and the allowed values are integers `[0, 100]`.

See LEF/DEF 5.8 Language Reference, Appendix C, "Calculating and Fixing Process Antenna Violations" for a *description* of antenna violations.

If no `diode_cell` argument is specified the LEF cell with class CORE, ANTENNACELL will be used. If any repairs are made the filler instances are remove and must be placed with the `filler_placement` command.

If the LEF technology layer `ANTENNADIFFSIDEAREARATIO` properties are constant instead of PWL, inserting diodes will not improve the antenna ratios, and thus, no diodes are inserted. The following warning message will be reported:

```
[WARNING GRT-0243] Unable to repair antennas on net with diodes.
```

Write Global Routing Guides

```
write_guides file_name
```

Switch Name	Description
file_name	Guide file name.

Example: `write_guides route.guide`.

Estimate Global Routing Parasitics

To estimate RC parasitics based on global route results, use the `-global_routing` option of the `estimate_parasitics` command.

Note: To see the function definition for `estimate_parasitics`, refer to *Resizer docs*.

```
estimate_parasitics -global_routing
```

Plot Global Routing Guides

The `draw_route_guides` command plots the route guides for a set of nets. To erase the route guides from the GUI, pass an empty list to this command: `draw_route_guides {}`.

```
draw_route_guides
    net_names
    [-show_pin_locations]
```

Options

Switch Name	Description
net_names	Tcl list of set of nets (e.g. {net1, net2}).
-show_pin_locations	Draw circles for the pin positions on the routing grid.

Report Wirelength

The `report_wire_length` command reports the wire length of the nets. Use the `-global_route` and the `-detailed_route` flags to report the wire length from global and detailed routing, respectively. If none of these flags are used, the tool will identify the state of the design and report the wire length accordingly.

```
report_wire_length
    [-net net_list]
    [-file file]
    [-global_route]
    [-detailed_route]
    [-verbose]
```

Options

Switch Name	Description
-net	List of nets to report the wirelength. Use * to report the wire length for all nets of the design.
-file	The name of the file for the wirelength report.
-global_route	Report the wire length of the global routing.
-detailed_route	Report the wire length of the detailed routing.
-verbose	This flag enables the full reporting of the layer-wise wirelength information.

Example: `report_wire_length -net {clk net60} -global_route -detailed_route -verbose -file out.csv`

Debug Mode

The `global_route_debug` command allows you to start a debug mode to view the status of the Steiner Trees. It also allows you to dump the input positions for the Steiner tree creation of a net. This must be used before calling the `global_route` command. Set the name of the net and the trees that you want to visualize.

```
global_route_debug
    [-st]
    [-rst]
    [-tree2D]
    [-tree3D]
    [-saveSttInput file_name]
    [-net net_name]
```

Options

Switch Name	Description
-st	Show the Steiner Tree generated by `stt`.
-rst	Show the Rectilinear Steiner Tree generated by `grt`.
-tree2D	Show the Rectilinear Steiner Tree generated by `grt` after the overflow iterations.
-tree3D	Show the 3D Rectilinear Steiner Tree post-layer assignment.
-saveSttInput	File name to save `stt` input of a net.
-net	The name of the net name to be displayed.

Example scripts

Examples scripts demonstrating how to run FastRoute on a sample design of `gcd` as follows:

```
./test/gcd.tcl
```

Read Global Routing Guides

```
read_guides file_name
```

Options

Switch Name	Description
file_name	Path to global routing guide.

Useful developer functions

If you are a developer, you might find these useful. More details can be found in the source file or the swig file.

Function Name	Description
check_routing_layer	Check if the layer is within the min/max routing layer specified.
parse_layer_name	Get routing layer number from layer name
parse_layer_range	Parses a range from layer_range argument of format (%s-%s). cmd argument is not used.
check_region	Checks the defined region if its within the die area.
define_layer_range	Provide a Tcl list of layers and automatically generate the min and max layers for signal routing.
`define_clock_layer_range	Provide a Tcl list of layers and automatically generate the min and max layers for clock routing.
have_detailed_route	Checks if block has detailed route already.

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

Using the Python interface to grt

> **Warning:** The Python interface is currently in development and is subject to change.

The Python API tries to stay close to the API defined in the C++ class GlobalRouter that is located here

When initializing a design, a sequence of Python commands might look like the following:

```
from openroad import Design, Tech
tech = Tech()
tech.readLef(...)
```

(continues on next page)

(continued from previous page)

```
design = Design(tech)
design.readDef(...)
gr = design.getGlobalRouter()
```

Here are some options to the `global_route` command. (See `GlobalRouter.h` for a complete list)

```
gr.setGridOrigin(x, y)                    # int, default 0,0
gr.setCongestionReportFile(file_name)     # string
gr.setOverflowIterations(n)               # int, default 50
gr.setAllowCongestion(allowCongestion)    # boolean, default False
gr.setCriticalNetsPercentage(percentage)  # float
gr.setMinRoutingLayer(minLayer)           # int
gr.setMaxRoutingLayer(maxLayer)           # int
gr.setMinLayerForClock(minLayer)          # int
gr.setMaxLayerForClock(maxLayer)          # int
gr.setVerbose(v)                          # boolean, default False
```

and when ready to actually do the global route:

```
gr.globalRoute(save_guides)               # boolean, default False
```

If you have set `save_guides` to True, you can then save the guides in `file_name` with:

```
design.getBlock().writeGuides(file_name)
```

You can find the index of a named layer with

```
lindex = tech.getDB().getTech().findLayer(layer_name)
```

or, if you only have the `Python` design object

```
lindex = design.getTech().getDB().getTech().findLayer(layer_name)
```

Be aware that much of the error checking is done in `Tcl`, so that with the current `C++` / `Python` API, that might be an issue to deal with. There are also some useful `Python` functions located in the `grt_aux.py` file but these are not considered a part of the *final* API and may be subject to change.

FAQs

Check out GitHub discussion about this tool.

References

- Database comes from OpenDB
- FastRoute 4.1 documentation. The FastRoute4.1 version was received from Yue Xu on June 15, 2019.
- Min Pan, Yue Xu, Yanheng Zhang and Chris Chu. “FastRoute: An Efficient and High-Quality Global Router. VLSI Design, Article ID 608362, 2012.” Available here.
- C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng and D. Karger, “Prim-Dijkstra Tradeoffs for Improved Performance-Driven Global Routing”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 14(7) (1995), pp. 890-896. Available here.

License

BSD 3-Clause License. See LICENSE file.

Flute3

Flute3 is an open-source rectilinear Steiner minimum tree heuristic with improvements made by UFRGS students and James Cherry. This tool is used for the calculation of wirelength in `grt` and `rsz`.

The version in this repository uses CMake and C++ namespace, and has dynamic memory allocation. Flute3 can handle nets with any degree.

External references (Optional)

The algorithm base is Flute3.1, extracted from the FastRoute4.1 version that was received from yuexu@iastate.edu on June 15, 2019, with the BSD-3 open source license as given in the FastRoute website.

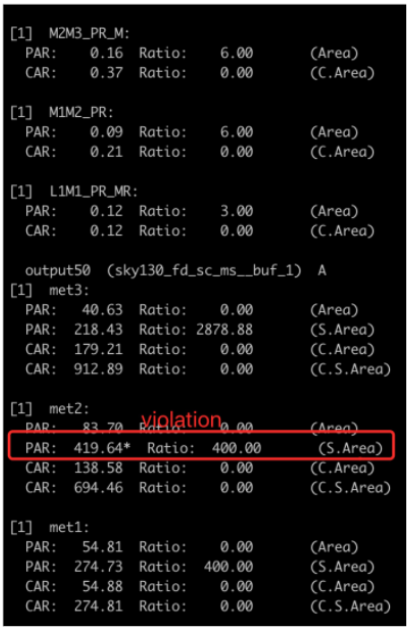
License

See LICENSE file.

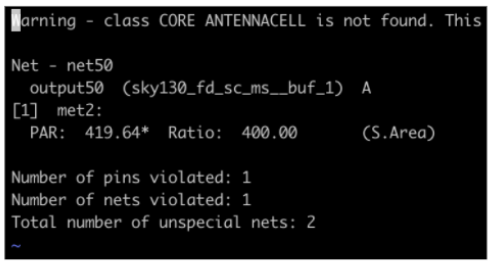
5.2.22 Antenna Rule Checker

This tool checks antenna violations and generates a report to indicate violated nets. See LEF/DEF 5.8 Language Reference, Appendix C, "Calculating and Fixing Process Antenna Violations" (p.389) for a description of antenna violations.

This is an example of the detailed and simple reports of the antenna checker:



Full report



Simple report

Abbreviations Index:

- `PAR`: Partial Area Ratio
- `CAR`: Cumulative Area Ratio
- `Area`: Gate Area
- `S. Area`: Side Diffusion Area
- `C. Area`: Cumulative Gate Area
- `C. S. Area`: Cumulative Side (Diffusion) Area

Antenna violations can be repaired after global routing with the `repair_design` command.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Check Antennas

```
check_antennas
    [-net net]
    [-verbose]
```

Options

Switch Name	Description
-verbose	Report all antenna calculations for violating nets.
-net	Check antennas on the specified net.

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

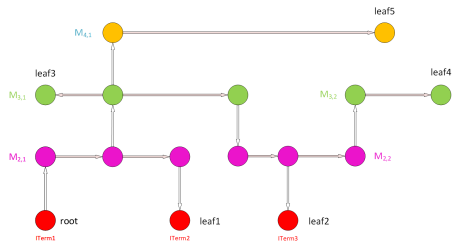
```
./test/regression
```

Limitations

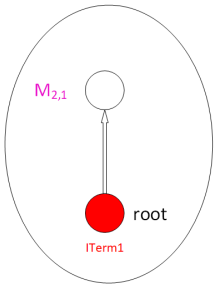
FAQs

Check out GitHub discussion about this tool.

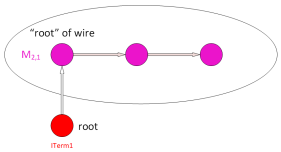
Algorithm



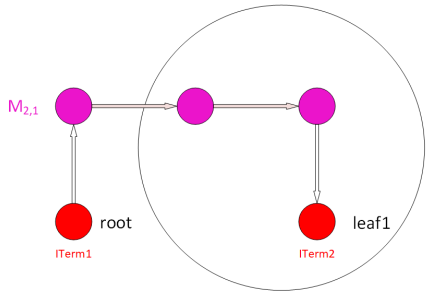
Antenna Checker Algorithm: WireGraph Example



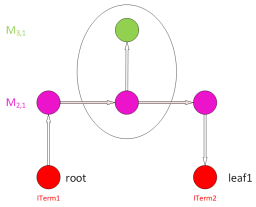
Step 1: (a) Start from the root node (ITerm) using upper Via to find a node for a new wire. (b) Save the ITerm area for cumulative gate/diffusion area.



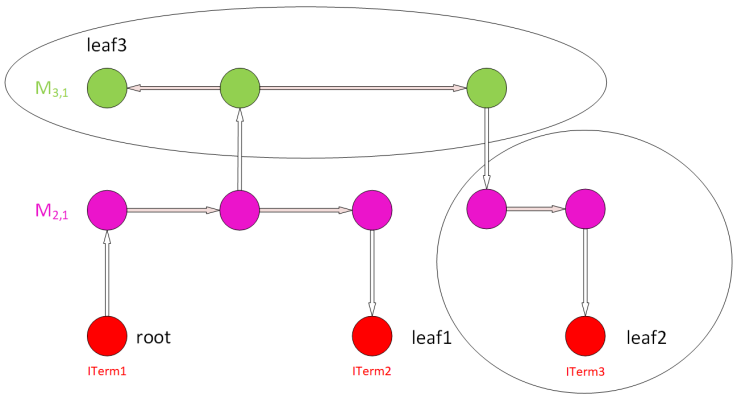
Step 2: From the node of the wire, find all the nodes in the wire through segment wires and find the "root" node of this wire.



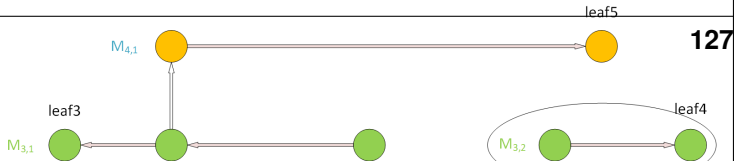
Step 3: (a) From the "root" node of the wire, along the outgoing segment edge that goes to other nodes belonging to this wire, calculate the area of this wire. (b) Then, find all the ITerms below these nodes, except for the root node (directly use an ITerm or lower Vias to find ITerms for lower metals). © Sum up the areas of all the ITerms found with the cumulative areas and calculate the PAR of this wire. (d) Add the PAR value and the wire info (layer, Index) into the PAR table. Add the new area to the cumulative areas.



Step 4: Find all the upper Vias on this wire (for all the nodes on this wire), and go to the higher-level metal.



Step 5: Repeat Steps 2 and 3 for new-found upper-level wires.



License

BSD 3-Clause License. See LICENSE file.

5.2.23 Detailed Routing

The Detailed Routing (`drt`) module in OpenROAD is based on the open-source detailed router, TritonRoute. TritonRoute consists of several main building blocks, including pin access analysis, track assignment, initial detailed routing, search and repair, and a DRC engine. The initial development of the router is inspired by the ISPD-2018 initial detailed routing contest. However, the current framework differs and is built from scratch, aiming for an industrial-oriented scalable and flexible flow.

TritonRoute provides industry-standard LEF/DEF interface with support of ISPD-2018 and ISPD-2019 contest-compatible route guide format.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Detailed Route

```
detailed_route
    [-output_maze filename]
    [-output_drc filename]
    [-output_cmap filename]
    [-output_guide_coverage filename]
    [-drc_report_iter_step step]
    [-db_process_node name]
    [-disable_via_gen]
    [-droute_end_iter iter]
    [-via_in_pin_bottom_layer layer]
    [-via_in_pin_top_layer layer]
    [-or_seed seed]
    [-or_k_ k]
    [-bottom_routing_layer layer]
    [-top_routing_layer layer]
    [-verbose level]
    [-distributed]
    [-remote_host rhost]
    [-remote_port rport]
    [-shared_volume vol]
    [-cloud_size sz]
    [-clean_patches]
    [-no_pin_access]
    [-min_access_points count]
```

(continues on next page)

(continued from previous page)

```
    [-save_guide_updates]
    [-repair_pdn_vias layer]
```

Options

Switch Name	Description
-output_maze	Path to output maze log file (e.g. output_maze.log).
-output_drc	Path to output DRC report file (e.g. output_drc.rpt).
-output_cmap	Path to output congestion map file (e.g. output.cmap).
-output_guide_coverage	Path to output guide coverage file (e.g. sample_coverage.csv).
-drc_report_iter_step	Report DRC on each iteration which is a multiple of this step. The default value is 0, and the allowed values are integers [0, MAX_INT].
-db_process_node	Specify the process node.
-disable_via_gen	Option to diable via generation with bottom and top routing layer. The default value is disabled.
-droute_end_iter	Number of detailed routing iterations. The default value is -1, and the allowed values are integers [1, 64].
-via_in_pin_bottom_layer	Via-in pin bottom layer name.
-via_in_pin_top_layer	Via-in pin top layer name.
-or_seed	Refer to developer arguments *here*.
-or_k	Refer to developer arguments *here*.
-bottom_routing_layer	Bottommost routing layer name.
-top_routing_layer	Topmost routing layer name.
-verbose	Sets verbose mode if the value is greater than 1, else non-verbose mode (must be integer, or error will be triggered.)
-distributed	Refer to distributed arguments *here*.
-clean_patches	Clean unneeded patches during detailed routing.
-no_pin_access	Disables pin access for routing.
-min_access_points	Minimum access points for standard cell and macro cell pins.
-save_guide_updates	Flag to save guides updates.
-repair_pdn_vias	This option is used for PDKs where M1 and M2 power rails run in parallel.

Developer arguments

Some arguments that are helpful for developers are listed here.

Switch Name	Description
-or_seed	Random seed for the order of nets to reroute. The default value is -1, and the allowed values are integers [0, MAX_INT].
-or_k	Number of swaps is given by $k * sizeof(rerouteNets)$. The default value is 0, and the allowed values are integers [0, MAX_INT].

Detailed Route Debugging

The following command and arguments are useful when debugging error messages from `drt` and to understand its behavior.

```
detailed_route_debug
    [-pa]
    [-ta]
    [-dr]
    [-maze]
    [-net name]
    [-pin name]
    [-worker x y]
    [-iter iter]
    [-pa_markers]
    [-dump_dr]
    [-dump_dir dir]
    [-pa_edge]
    [-pa_commit]
    [-write_net_tracks]
```

Options

Switch Name	Description
`-pa`	Enable debug for pin access.
`-ta`	Enable debug for track assignment.
`-dr`	Enable debug for detailed routing.
`-maze`	Enable debug for maze routing.
`-net`	Enable debug for net name.
`-pin`	Enable debug for pin name.
`-worker`	Debugs routes that pass through the point `{x, y}`.
`-iter`	Specifies the number of debug iterations. The default value is `0`, and the accepted values are integers `[0, MAX_INT`.
`-pa_markers`	Enable pin access markers.
`-dump_dr`	Filename for detailed routing dump.
`-dump_dir`	Directory for detailed routing dump.
`-pa_edge`	Enable visibility of pin access edges.
`-pa_commit`	Enable visibility of pin access commits.
`-write_net_tracks`	Enable writing of net track assigments.

Check Pin Access

```
pin_access
    [-db_process_node name]
    [-bottom_routing_layer layer]
    [-top_routing_layer layer]
    [-min_access_points count]
    [-verbose level]
    [-distributed]
```

(continues on next page)

(continued from previous page)

```
    [-remote_host rhost]
    [-remote_port rport]
    [-shared_volume vol]
    [-cloud_size sz]
```

Options

Switch Name	Description
-db_process_node	Specify process node.
-bottom_routing_layer	Bottommost routing layer.
-top_routing_layer	Topmost routing layer.
-min_access_points	Minimum number of access points per pin.
-verbose	Sets verbose mode if the value is greater than 1, else non-verbose mode (must be integer, or error will be triggered.)
-distributed	Refer to distributed arguments *here*.

Distributed arguments

We have compiled all distributed arguments in this section.

Note: Additional setup is required. Please refer to this guide.

Switch Name	Description
-distributed	Enable distributed mode with Kubernetes and Google Cloud.
-remote_host	The host IP.
-remote_port	The value of the port to access from.
-shared_volume	The mount path of the nfs shared folder.
-cloud_size	The number of workers.

Useful developer functions

If you are a developer, you might find these useful. More details can be found in the source file or the swig file.

Function Name	Description
detailed_route_set_default_via	Set default via.
detailed_route_set_unidirectional_layer	Set unidirectional layer.
step_dr	Refer to function detailed_route_step_drt.
check_drc	Refer to function check_drc_cmd.

Example scripts

Example script demonstrating how to run TritonRoute on a sample design of `gcd` in the Nangate45 technology node.

```
# single machine example
./test/gcd_nangate45.tcl

# distributed example
./test/gcd_nangate45_distributed.tcl
```

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

References

Please cite the following paper(s) for publication:

1. A. B. Kahng, L. Wang and B. Xu, "TritonRoute: The Open Source Detailed Router", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2020), doi:10.1109/TCAD.2020.3003234. (.pdf)
2. A. B. Kahng, L. Wang and B. Xu, "The Tao of PAO: Anatomy of a Pin Access Oracle for Detailed Routing", Proc. ACM/IEEE Design Automation Conf., 2020, pp. 1-6. (.pdf)

Authors

TritonRoute was developed by graduate students Lutong Wang and Bangqi Xu from UC San Diego, and serves as the detailed router in the OpenROAD project.

License

5.2.24 Metal fill

This module inserts floating metal fill shapes to meet metal density design rules while obeying DRC constraints. It is driven by a `json` configuration file.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Density Fill

```
density_fill
    [-rules rules_file]
    [-area {lx ly ux uy}]
```

Options

Switch Name	Description
`-rules`	Specify `json` rule file.
`-area`	Optional. If not specified, the core area will be used.

Example scripts

The rules `json` file controls fill and you can see an example here.

The schema for the `json` is:

```
{
  "layers": {
    "<group_name>": {
      "layers": "<list of integer gds layers>",
      "names": "<list of name strings>",
      "opc": {
        "datatype":  "<list of integer gds datatypes>",
        "width":   "<list of widths in microns>",
        "height":   "<list of heightsin microns>",
        "space_to_fill": "<real: spacing between fills in microns>",
        "space_to_non_fill": "<real: spacing to non-fill shapes in microns>",
        "space_line_end": "<real: spacing to end of line in microns>"
      },
      "non-opc": {
        "datatype":  "<list of integer gds datatypes>",
        "width":   "<list of widths in microns>",
```

(continues on next page)

(continued from previous page)

```
        "height":   "<list of heightsin microns>",
        "space_to_fill": "<real: spacing between fills in microns>",
        "space_to_non_fill": "<real: spacing to non-fill shapes in microns>"
      }
    }, ...
  }
}
```

The `opc` section is optional depending on your process.

The width/height lists are effectively parallel arrays of shapes to try in left to right order (generally larger to smaller).

The layer grouping is for convenience. For example in some technologies many layers have similar rules so it is convenient to have a `Mx`, `Cx` group.

This all started out in `klayout` so there are some obsolete fields that the parser accepts but ignores (e.g., `space_to_outline`).

Regression tests

There are a set of regression tests in `./test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Limitations

FAQs

Check out GitHub discussion about this tool.

License

BSD 3-Clause License. See LICENSE file.

5.2.25 Parasitics Extraction

The parasitics extraction module in OpenROAD (`rcx`) is based on the open-source OpenRCX, a Parasitic Extraction (PEX, or RCX) tool that works on OpenDB design APIs. It extracts routed designs based on the LEF/DEF layout model.

OpenRCX extracts both Resistance and Capacitance for wires, based on coupling distance to the nearest wire and the track density context over and/or under the wire of interest, as well as cell abstracts. The capacitance and resistance measurements are based on equations of coupling distance interpolated on exact measurements from a calibration file, called the Extraction Rules file. The Extraction Rules file (RC technology file) is generated once for every process node and corner, using a provided utility for DEF wire pattern generation and regression modeling.

OpenRCX stores resistance, coupling capacitance and ground (i.e., grounded) capacitance on OpenDB objects with direct pointers to the associated wire and via db objects. Optionally, OpenRCX can generate a `.spef` file.

Commands

Note:

- Parameters in square brackets `[-param param]` are optional.
- Parameters without square brackets `-param2 param2` are required.

Define Process Corner

```
define_process_corner
    [-ext_model_index index]
    filename
```

Options

Switch Name	Description
`-ext_model_index`	Extraction model index. Expects 2 inputs (an index, and corner name).
`filename`	Path to process corner file `rcx_patterns.rules`.

Extract Parasitics

The `extract_parasitics` command performs parasitic extraction based on the routed design. If there are no information on routed design, no parasitics are returned.

```
extract_parasitics
    [-ext_model_file filename]
    [-corner_cnt count]
    [-max_res ohms]
    [-coupling_threshold fF]
    [-debug_net_id id]
    [-lef_res]
    [-cc_model track]
    [-context_depth depth]
    [-no_merge_via_res]
```

Options

Switch Name	Description
`-ext_model_file`	Specify the Extraction Rules file used for the extraction.
`-corner_cnt`	Defines the number of corners used during the parasitic extraction.
`-max_res`	Combines resistors in series up to the threshold value.
`-coupling_threshold`	Coupling below this threshold is grounded. The default value is `0.1`, units are in `fF`, accepted values are floats.
`-debug_net_id`	*Developer Option*: Net ID to evaluate.
`-lef_res`	Override LEF resistance per unit.
`-cc_model`	Specify the maximum number of tracks of lateral context that the tool considers on the same routing level. The default value is `10`, and the allowed values are integers `[0, MAX_INT]`.
`-context_depth`	Specify the number of levels of vertical context that OpenRCX needs to consider for the over/under context overlap for capacitance calculation. The default value is `5`, and the allowed values are integers `[0, MAX_INT]`.
`-no_merge_via_res`	Separates the via resistance from the wire resistance.

Write SPEF

The `write_spef` command writes the `.spef` output of the parasitics stored in the database.

```
write_spef
    [-net_id net_id]
    [-nets nets]
    filename
```

Options

Switch Name	Description
`-net_id`	Output the parasitics info for specific net IDs.
`-nets`	Net name.
`filename`	Output filename.

Scale RC

Use the `adjust_rc` command to scale the resistance, ground, and coupling capacitance.

```
adjust_rc
    [-res_factor res]
    [-cc_factor cc]
    [-gndc_factor gndc]
```

Options

Switch Name	Description
-res_factor	Scale factor for resistance.
-cc_factor	Scale factor for coupling capacitance.
-gndc_factor	Scale factor for ground capacitance.

Comparing SPEF files

The diff_spef command compares the parasitics in the reference database <filename>.spef. The output of this command is diff_spef.out and contains the RC numbers from the parasitics in the database and the <filename>.spef, and the percentage RC difference of the two data.

```
diff_spef
    [-file filename]
    [-r_res]
    [-r_cap]
    [-r_cc_cap]
    [-r_conn]
```

Options

Switch Name	Description
-file	Path to the input .spef filename.
-r_res	Read resistance.
-r_cap	Read capacitance.
-r_cc_cap	Read coupled capacitance.
r_conn	Read connections.

Extraction Rules File Generation

The bench_wires command produces a layout which contains various patterns that are used to characterize per-unit length R and C values. The generated patterns model the lateral, vertical, and diagonal coupling capacitances, as well as ground capacitance effects. This command generates a .def file that contains a number of wire patterns.

This command is specifically intended for the Extraction Rules file generation only.

```
bench_wires
    [-met_cnt mcnt]
    [-cnt count]
    [-len wire_len]
    [-over]
    [-diag]
    [-all]
    [-db_only]
    [-under_met layer]
    [-w_list width]
    [-s_list space]
```

(continues on next page)

(continued from previous page)

```
    [-over_dist dist]
    [-under_dist dist]
```

Options

Switch Name	Description
-met_cnt	Number of layers used in each pattern. The default value is -1, meaning it is not set, and the allowed values are integers [0, MAX_INT].
-cnt	Number of wires in each pattern. The default value is 5, and the default values are integers [0, MAX_INT].
-len	Wirelength in microns in the pattern. The default value is 100, and the allowed values are integers [0, MAX_INT].
-all	Consider all different pattern geometries (over, under, over_under, and diagonal).
-db_only	Run with db values only. All parameters in bench_wires are ignored.
-under_met	Consider under metal layer.
-w_list	Lists of wire width multipliers from the minimum spacing defined in the LEF.
-s_list	Lists of wire spacing multipliers from the minimum spacing defined in the LEF. The list will be the input index on the OpenRCX RC table (Extraction Rules file).
-over_dist, -under_dist	Consider over and under metal distance respectively.

Generate verilog netlist

bench_verilog is used after the bench_wires command. This command generates a Verilog netlist of the generated pattern layout by the bench_wires command.

This command is optional when running the Extraction Rules generation flow. This step is required if the favorite extraction tool (i.e., reference extractor) requires a Verilog netlist to extract parasitics of the pattern layout.

```
bench_verilog
    [filename]
```

Options

Switch Name	Description
`filename`	Name for the Verilog output file (e.g., `output.v`).

Read SPEF

The `bench_read_spef` command reads a <`filename`>`.spef` file and stores the parasitics into the database.

```
bench_read_spef
    [filename]
```

Options

Switch Name	Description
`filename`	Path to the input `.spef` file.

Write Rule File

The `write_rules` command writes the Extraction Rules file (RC technology file) for OpenRCX. It processes the parasitics data from the layout patterns that are generated using the `bench_wires` command, and writes the Extraction Rules file with <`filename`> as the output file.

This command is specifically intended for the purpose of Extraction Rules file generation.

```
write_rules
  [-file filename]
  [-dir dir]
  [-name name]
  [-pattern pattern]
```

Options

Switch Name	Description
`-file`	Output file name.
`-dir`	Output file directory.
`-name`	Name of rule.
`-pattern`	Flag to write the pattern to rulefile (0/1).

Example scripts

Example scripts demonstrating how to run OpenRCX in the OpenROAD environment on sample designs can be found in /test. An example flow test taking a sample design from synthesizable RTL Verilog to final-routed layout in an open-source SKY130 technology is shown below.

```
./test/gcd.tcl
```

Example scripts demonstrating how to run the Extraction Rules file generation can be found in this directory.

```
./calibration/script/generate_patterns.tcl      # generate patterns
./calibration/script/generate_rules.tcl         # generate the Extraction Rules file
./calibration/script/ext_patterns.tcl           # check the accuracy of OpenRCX
```

Regression tests

There are a set of regression tests in `/test`. For more information, refer to this *section*.

Simply run the following script:

```
./test/regression
```

Extraction Rules File Generation

This flow generates an Extraction Rules file (RC tech file, or RC table) for OpenRCX. This file provides resistance and capacitance tables used for RC extraction for a specific process corner.

The Extraction Rules file (RC technology file) is generated once for every process node and corner automatically.

The detailed documentation can be found *here*.

Limitations

FAQs

Check out GitHub discussion about this tool.

License

BSD 3-Clause License. See LICENSE file.

Extraction Rules Generation Flow for OpenRCX

This flow generates the RC tech file for OpenRCX. The RC tech file provides resistance and capacitance tables used for RC extraction for a specific process corner.

The flow involves:

A. Running OpenRCX `generate_patterns.tcl` to generate layout patterns.

- Input: tech LEF
- Output: `patterns.def`, `patterns.v`
- Script: `generate_patterns.tcl`
- Desc: OpenRCX generates many pattern geometries to model various types of capacitance and resistance (i.e., multi-conductor) geometric configurations.

B. Running your favorite extraction tool (i.e., reference extractor) to extract parasitics of the layout patterns.

- Input: `patterns.def`, `patterns.v` (if required), and additional files required by the reference extractor.
- Output: `patterns.spef`
- Script: Not provided
- Desc: Extract parasitics of the patterns generated by OpenRCX using a reference extractor. This one-time step provides the parasitics of various types of pattern geometries as reference for fitted per-unit length R, C calculation.

C. Running OpenRCX to convert `patterns.spef` to RC tech file.

- Input: `patterns.spef`
- Output: RC tech file
- Script: `generate_rules.tcl`
- Desc: OpenRCX takes the `.spef` from the reference extractor and performs calculations to produce capacitance and resistance tables for a wide range of wire geometries. The output of this flow is a custom RC tech file for OpenRCX.

D. Benchmarking - test the accuracy of OpenRCX on the patterns layout.

- Input: `patterns.def` and RC tech file
- Output: `rcx.spef`, `diff_spef.out`
- Script: `ext_patterns.tcl`
- Desc: Perform parasitic extraction on pattern layout for the calibration using the generated RC tech file. OpenRCX then compares the extracted parasitics with the golden parasitics that had been extracted by the reference extractor in Step (B) above.

How to run:

1. Go to OpenRCX home directory (`./OpenROAD/src/rcx`).
2. Navigate to calibration folder `cd calibration`
3. Modify the `user_env.tcl` script in the script directory.

- TECH_LEF: points to the directory of the tech LEF
- PROCESS_NODE: the technology node
- extRules: the name and the location of the OpenRCX tech file

4. Run the executable script `run.sh` –> run Steps (A) through (D) of the flow above.

- `source run.sh` or `./run.sh`

5. The OpenRCX RC tech file can be found in the directory that is specified in the extRules variable.

5.2.26 OpenROAD Messages Glossary

Listed below are the OpenROAD warning/errors you may encounter while using the application.

Tool	Code	Filename:Line Number	Type	Information
ANT	0001	AntennaChecker.cc:1776	INFO	-
ANT	0002	AntennaChecker.cc:1774	INFO	-
ANT	0008	AntennaChecker.cc:1733	ERROR	-
ANT	0009	AntennaChecker.cc:1996	WARN	-
ANT	0010	AntennaChecker.tcl:56	WARN	-
ANT	0011	AntennaChecker.tcl:59	WARN	-
ANT	0012	AntennaChecker.i:66	ERROR	-
ANT	0013	AntennaChecker.cc:205	WARN	-
ANT	0014	AntennaChecker.cc:1758	ERROR	-
CTS	0001	TritonCTS.cpp:147	INFO	-
CTS	0003	TritonCTS.cpp:375	INFO	-
CTS	0004	TritonCTS.cpp:379	INFO	-
CTS	0005	TritonCTS.cpp:383	INFO	-
CTS	0006	TritonCTS.cpp:387	INFO	-
CTS	0007	TritonCTS.cpp:593	INFO	-
CTS	0008	TritonCTS.cpp:618	INFO	-
CTS	0010	TritonCTS.cpp:698	INFO	-
CTS	0012	TritonCTS.cpp:846	INFO	-
CTS	0013	TritonCTS.cpp:848	INFO	-
CTS	0014	TritonCTS.cpp:852	INFO	-
CTS	0015	TritonCTS.cpp:856	INFO	-
CTS	0016	TritonCTS.cpp:865	INFO	-
CTS	0017	TritonCTS.cpp:869	INFO	-
CTS	0018	TritonCTS.cpp:1019	INFO	-
CTS	0019	HTreeBuilder.cpp:271	INFO	-
CTS	0020	HTreeBuilder.cpp:283	INFO	-
CTS	0021	HTreeBuilder.cpp:292	INFO	-
CTS	0022	HTreeBuilder.cpp:300	INFO	-
CTS	0023	HTreeBuilder.cpp:327	INFO	-
CTS	0024	HTreeBuilder.cpp:344	INFO	-
CTS	0025	HTreeBuilder.cpp:345	INFO	-
CTS	0026	HTreeBuilder.cpp:346	INFO	-
CTS	0027	HTreeBuilder.cpp:1125	INFO	-
CTS	0028	HTreeBuilder.cpp:1127	INFO	-
CTS	0029	HTreeBuilder.cpp:1133	INFO	-
CTS	0030	HTreeBuilder.cpp:1141	INFO	-
CTS	0031	HTreeBuilder.cpp:1163	INFO	-
CTS	0032	HTreeBuilder.cpp:1175	INFO	-
CTS	0034	HTreeBuilder.cpp:1329	INFO	-
CTS	0035	HTreeBuilder.cpp:1900	INFO	-
CTS	0038	TechChar.cpp:1162	INFO	-
CTS	0039	TechChar.cpp:1179	INFO	-
CTS	0040	TritonCTS.cpp:605	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
CTS	0041	TritonCTS.cpp:690	WARN	-
CTS	0043	TechChar.cpp:135	WARN	-
CTS	0045	TechChar.cpp:382	WARN	-
CTS	0046	TechChar.cpp:142	INFO	-
CTS	0047	TechChar.cpp:144	INFO	-
CTS	0048	TechChar.cpp:149	INFO	-
CTS	0049	TechChar.cpp:504	INFO	-
CTS	0056	TritonCTS.tcl:166	ERROR	-
CTS	0058	SinkClustering.cpp:174	ERROR	-
CTS	0065	TechChar.cpp:178	ERROR	-
CTS	0073	TechChar.cpp:456	ERROR	-
CTS	0074	TechChar.cpp:461	ERROR	-
CTS	0075	TechChar.cpp:566	ERROR	-
CTS	0076	TechChar.cpp:610	ERROR	-
CTS	0078	TechChar.cpp:637	ERROR	-
CTS	0079	HTreeBuilder.cpp:200	ERROR	-
CTS	0080	HTreeBuilder.cpp:1892	ERROR	-
CTS	0081	TritonCTS.cpp:132	ERROR	-
CTS	0082	TritonCTS.cpp:97	WARN	-
CTS	0083	TritonCTS.cpp:615	WARN	-
CTS	0084	TechChar.cpp:80	INFO	-
CTS	0085	TritonCTS.cpp:827	ERROR	-
CTS	0087	TritonCTS.cpp:361	ERROR	-
CTS	0090	HTreeBuilder.cpp:1130	INFO	-
CTS	0093	LevelBalancer.cpp:52	INFO	-
CTS	0095	TritonCTS.cpp:591	INFO	-
CTS	0096	TechChar.cpp:587	ERROR	-
CTS	0097	TritonCTS.cpp:138	INFO	-
CTS	0098	TritonCTS.cpp:336	INFO	-
CTS	0099	TritonCTS.cpp:337	INFO	-
CTS	0100	TritonCTS.cpp:338	INFO	-
CTS	0101	TritonCTS.cpp:340	INFO	-
CTS	0102	TritonCTS.cpp:341	INFO	-
CTS	0103	TritonCTS.tcl:209	ERROR	-
CTS	0104	TechChar.cpp:436	WARN	-
CTS	0105	TritonCTS.cpp:670	WARN	-
CTS	0106	TechChar.cpp:468	ERROR	-
CTS	0107	TechChar.cpp:597	ERROR	-
CTS	0108	TechChar.cpp:604	ERROR	-
CTS	0111	TechChar.cpp:488	ERROR	-
CTS	0113	TechChar.cpp:498	ERROR	-
CTS	0114	TritonCTS.cpp:575	ERROR	-
CTS	0115	TritonCTS.tcl:103	WARN	-
CTS	0116	TritonCTS.cpp:181	INFO	-
CTS	0200	TreeBuilder.cpp:56	INFO	-
CTS	0201	TreeBuilder.cpp:71	INFO	-
CTS	0202	TritonCTS.cpp:921	INFO	-
CTS	0203	TritonCTS.cpp:1134	WARN	-
CTS	0204	HTreeBuilder.cpp:160	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
CTS	0205	HTreeBuilder.cpp:167	INFO	-
CTS	0206	HTreeBuilder.cpp:173	INFO	-
CTS	0534	TechChar.cpp:522	ERROR	-
CTS	0541	TechChar.cpp:527	ERROR	-
CTS	0542	CtsOptions.h:162	ERROR	-
CTS	0543	CtsOptions.h:178	ERROR	-
DFT	0002	ScanReplace.cpp:384	WARN	-
DFT	0003	ScanReplace.cpp:374	INFO	-
DFT	0004	ClockDomain.cpp:52	ERROR	-
DFT	0005	ScanCellFactory.cpp:137	WARN	-
DFT	0006	dft.i:80	ERROR	-
DFT	0007	ScanCellFactory.cpp:146	WARN	-
DPL	0001	FillerPlacement.cpp:73	INFO	-
DPL	0002	FillerPlacement.cpp:110	ERROR	-
DPL	0012	dbToOpendp.cpp:131	ERROR	-
DPL	0013	Grid.cpp:633	ERROR	-
DPL	0015	Place.cpp:315	ERROR	-
DPL	0016	Place.cpp:427	ERROR	-
DPL	0017	Place.cpp:481	ERROR	-
DPL	0020	OptMirror.cpp:98	INFO	-
DPL	0021	OptMirror.cpp:100	INFO	-
DPL	0022	OptMirror.cpp:102	INFO	-
DPL	0023	OptMirror.cpp:107	INFO	-
DPL	0026	Place.cpp:1318	CRITICAL	-
DPL	0027	Opendp.tcl:77	ERROR	-
DPL	0028	Opendp.tcl:210	WARN	-
DPL	0029	Opendp.tcl:227	ERROR	-
DPL	0030	Opendp.tcl:242	ERROR	-
DPL	0031	Opendp.tcl:53	ERROR	-
DPL	0032	Opendp.tcl:185	ERROR	-
DPL	0033	CheckPlacement.cpp:129	ERROR	-
DPL	0034	Opendp.cpp:172	INFO	-
DPL	0035	Opendp.cpp:177	INFO	-
DPL	0036	Opendp.cpp:184	ERROR	-
DPL	0037	Opendp.cpp:144	WARN	-
DPL	0038	Opendp.cpp:160	WARN	-
DPL	0039	*Opendp.tcl:214*	any	-
This	could	be	ERROR	-
DPL	0040	CheckPlacement.cpp:210	ERROR	-
DPL	0041	Grid.cpp:680	ERROR	-
DPL	0042	Grid.cpp:508	WARN	-
DPL	0043	Opendp.cpp:577	ERROR	-
DPL	0044	Opendp.cpp:582	ERROR	-
DPL	0045	CheckPlacement.cpp:272	ERROR	-
DPL	0048	CheckPlacement.cpp:151	ERROR	-
DPO	0001	detailed.cxx:174	ERROR	-
DPO	0031	Optdp.tcl:59	ERROR	-
DPO	0100	Optdp.cpp:431	INFO	-
DPO	0101	Optdp.cpp:560	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
DPO	0102	Optdp.cpp:594	ERROR	-
DPO	0103	Optdp.cpp:622	ERROR	-
DPO	0104	Optdp.cpp:635	ERROR	-
DPO	0105	Optdp.cpp:651	ERROR	-
DPO	0106	Optdp.cpp:661	ERROR	-
DPO	0107	Optdp.cpp:668	ERROR	-
DPO	0108	Optdp.cpp:750	WARN	-
DPO	0109	Optdp.cpp:675	INFO	-
DPO	0110	Optdp.cpp:929	INFO	-
DPO	0200	legalize_shift.cxx:198	WARN	-
DPO	0201	legalize_shift.cxx:214	WARN	-
DPO	0202	detailed_mis.cxx:185	INFO	-
DPO	0203	detailed_random.cxx:398	INFO	-
DPO	0300	detailed_mis.cxx:164	INFO	-
DPO	0301	detailed_mis.cxx:194	INFO	-
DPO	0302	detailed_mis.cxx:226	INFO	-
DPO	0303	detailed.cxx:176	INFO	-
DPO	0304	detailed_reorder.cxx:113	INFO	-
DPO	0305	detailed_reorder.cxx:127	INFO	-
DPO	0306	detailed_global.cxx:133	INFO	-
DPO	0307	detailed_global.cxx:141	INFO	-
DPO	0308	detailed_vertical.cxx:135	INFO	-
DPO	0309	detailed_vertical.cxx:147	INFO	-
DPO	0310	detailed_manager.cxx:949	INFO	-
DPO	0311	detailed_manager.cxx:1351	INFO	-
DPO	0312	detailed_manager.cxx:1406	INFO	-
DPO	0313	detailed_manager.cxx:1587	INFO	-
DPO	0314	detailed_manager.cxx:1645	INFO	-
DPO	0315	detailed_manager.cxx:1677	INFO	-
DPO	0317	detailed_abu.cxx:422	INFO	-
DPO	0318	detailed_manager.cxx:1187	INFO	-
DPO	0319	detailed_manager.cxx:1231	INFO	-
DPO	0320	detailed_manager.cxx:1261	INFO	-
DPO	0321	detailed_manager.cxx:1293	INFO	-
DPO	0322	detailed_manager.cxx:303	INFO	-
DPO	0323	detailed.cxx:131	WARN	-
DPO	0324	detailed_random.cxx:192	INFO	-
DPO	0325	detailed_random.cxx:242	INFO	-
DPO	0326	detailed_random.cxx:264	INFO	-
DPO	0327	detailed_random.cxx:302	INFO	-
DPO	0328	detailed_random.cxx:321	INFO	-
DPO	0329	detailed_random.cxx:390	INFO	-
DPO	0330	detailed_random.cxx:428	INFO	-
DPO	0332	detailed_random.cxx:501	INFO	-
DPO	0333	detailed_random.cxx:515	INFO	-
DPO	0334	detailed_global.cxx:542	INFO	-
DPO	0335	detailed_random.cxx:671	INFO	-
DPO	0336	detailed_vertical.cxx:535	INFO	-
DPO	0337	detailed_random.cxx:843	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
DPO	0338	detailed_random.cxx:534	INFO	-
DPO	0339	detailed_manager.cxx:1467	WARN	-
DPO	0340	detailed_manager.cxx:1477	WARN	-
DPO	0380	detailed_orient.cxx:104	INFO	-
DPO	0381	detailed_orient.cxx:112	WARN	-
DPO	0382	detailed_orient.cxx:118	INFO	-
DPO	0383	detailed_orient.cxx:127	INFO	-
DPO	0384	detailed_orient.cxx:132	INFO	-
DPO	0385	detailed_random.cxx:589	ERROR	-
DPO	0400	detailed_manager.cxx:171	ERROR	-
DPO	0401	detailed_manager.cxx:134	INFO	-
DPO	0402	detailed_manager.cxx:152	INFO	-
DRT	0000	FlexDR_init.cpp:331	ERROR	-
DRT	0001	TritonRoute.cpp:1021	ERROR	-
DRT	0002	TritonRoute.cpp:216	ERROR	-
DRT	0003	io.cpp:3373	ERROR	-
DRT	0004	io.cpp:3379	ERROR	-
DRT	0005	frRegionQuery.cpp:118	ERROR	-
DRT	0006	frRegionQuery.cpp:129	ERROR	-
DRT	0007	frRegionQuery.cpp:147	ERROR	-
DRT	0008	frRegionQuery.cpp:315	ERROR	-
DRT	0009	frRegionQuery.cpp:341	ERROR	-
DRT	0010	frRegionQuery.cpp:351	ERROR	-
DRT	0011	frRegionQuery.cpp:375	ERROR	-
DRT	0012	frRegionQuery.cpp:386	ERROR	-
DRT	0013	frRegionQuery.cpp:397	ERROR	-
DRT	0014	frRegionQuery.cpp:445	ERROR	-
DRT	0015	frRegionQuery.cpp:462	ERROR	-
DRT	0016	frRegionQuery.cpp:507	ERROR	-
DRT	0017	frRegionQuery.cpp:526	ERROR	-
DRT	0018	frRegionQuery.cpp:729	INFO	-
DRT	0019	frRegionQuery.cpp:733	INFO	-
DRT	0020	frRegionQuery.cpp:745	INFO	-
DRT	0021	frRegionQuery.cpp:749	INFO	-
DRT	0022	frRegionQuery.cpp:766	INFO	-
DRT	0023	frRegionQuery.cpp:777	INFO	-
DRT	0024	frRegionQuery.cpp:787	INFO	-
DRT	0026	frRegionQuery.cpp:817	INFO	-
DRT	0027	frRegionQuery.cpp:821	INFO	-
DRT	0028	frRegionQuery.cpp:832	INFO	-
DRT	0029	frRegionQuery.cpp:861	INFO	-
DRT	0030	frRegionQuery.cpp:865	INFO	-
DRT	0031	frRegionQuery.cpp:158	ERROR	-
DRT	0032	frRegionQuery.cpp:993	INFO	-
DRT	0033	frRegionQuery.cpp:1007	INFO	-
DRT	0034	frRegionQuery.cpp:1035	INFO	-
DRT	0035	frRegionQuery.cpp:875	INFO	-
DRT	0036	frRegionQuery.cpp:1021	INFO	-
DRT	0037	FlexGC_init.cpp:89	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
DRT	0038	FlexGC_init.cpp:100	ERROR	-
DRT	0039	FlexGC_init.cpp:106	ERROR	-
DRT	0041	FlexGC_main.cpp:130	WARN	-
DRT	0042	FlexGC_main.cpp:205	ERROR	-
DRT	0043	FlexGC_main.cpp:284	WARN	-
DRT	0044	FlexGC_main.cpp:2413	WARN	-
DRT	0045	FlexGC_main.cpp:2592	WARN	-
DRT	0046	FlexGC_main.cpp:2599	WARN	-
DRT	0047	FlexGC_main.cpp:2606	WARN	-
DRT	0048	FlexGC_main.cpp:2613	WARN	-
DRT	0050	FlexGC_main.cpp:2620	WARN	-
DRT	0051	FlexGC_main.cpp:2627	WARN	-
DRT	0052	FlexGC_main.cpp:2634	WARN	-
DRT	0053	FlexGC_init.cpp:985	ERROR	-
DRT	0054	FlexGC_main.cpp:2725	WARN	-
DRT	0055	FlexGC_main.cpp:2729	WARN	-
DRT	0056	FlexGC_main.cpp:2737	WARN	-
DRT	0057	FlexGC_main.cpp:2744	WARN	-
DRT	0058	FlexGC_main.cpp:2749	WARN	-
DRT	0059	FlexGC_main.cpp:2754	WARN	-
DRT	0060	FlexGC_main.cpp:2760	WARN	-
DRT	0061	FlexGC_main.cpp:2767	WARN	-
DRT	0062	FlexGC_main.cpp:2772	WARN	-
DRT	0063	FlexGC_main.cpp:2777	WARN	-
DRT	0065	FlexPA_unique.cpp:89	ERROR	-
DRT	0066	FlexPA_unique.cpp:98	WARN	-
DRT	0067	FlexPA_prep.cpp:117	ERROR	-
DRT	0068	FlexPA_prep.cpp:458	ERROR	-
DRT	0069	FlexPA_unique.cpp:253	ERROR	-
DRT	0070	FlexPA_prep.cpp:801	ERROR	-
DRT	0073	FlexPA_prep.cpp:1548	ERROR	-
DRT	0074	FlexPA_prep.cpp:1595	ERROR	-
DRT	0076	FlexPA_prep.cpp:1560	INFO	-
DRT	0077	FlexPA_prep.cpp:1564	INFO	-
DRT	0078	FlexPA_prep.cpp:1606	INFO	-
DRT	0079	FlexPA_prep.cpp:1771	INFO	-
DRT	0080	FlexPA_prep.cpp:1776	INFO	-
DRT	0081	FlexPA_prep.cpp:1788	INFO	-
DRT	0082	FlexPA_prep.cpp:1701	INFO	-
DRT	0083	FlexPA_prep.cpp:1705	INFO	-
DRT	0084	FlexPA_prep.cpp:1717	INFO	-
DRT	0085	FlexPA_prep.cpp:2015	ERROR	-
DRT	0086	FlexPA_prep.cpp:2271	ERROR	-
DRT	0087	FlexPA_prep.cpp:1757	WARN	-
DRT	0089	FlexPA_prep.cpp:2454	WARN	-
DRT	0090	FlexPA_prep.cpp:2699	ERROR	-
DRT	0091	FlexPA_prep.cpp:2772	ERROR	-
DRT	0092	FlexRP_prep.cpp:1510	WARN	-
DRT	0093	FlexRP_prep.cpp:1534	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
DRT	0094	io.cpp:91	ERROR	-
DRT	0095	io.cpp:133	ERROR	-
DRT	0096	io.cpp:137	ERROR	-
DRT	0097	io.cpp:216	ERROR	-
DRT	0098	io.cpp:226	ERROR	-
DRT	0099	io.cpp:247	ERROR	-
DRT	0100	io.cpp:339	ERROR	-
DRT	0101	io.cpp:341	ERROR	-
DRT	0102	io.cpp:480	ERROR	-
DRT	0103	io.cpp:509	ERROR	-
DRT	0104	io.cpp:554	ERROR	-
DRT	0105	io.cpp:579	ERROR	-
DRT	0106	io.cpp:585	ERROR	-
DRT	0107	io.cpp:675	ERROR	-
DRT	0108	io.cpp:808	ERROR	-
DRT	0109	io.cpp:871	ERROR	-
DRT	0110	FlexPA_prep.cpp:1656	INFO	-
DRT	0111	FlexPA_prep.cpp:1660	INFO	-
DRT	0112	io.cpp:996	ERROR	-
DRT	0113	io.cpp:3067	ERROR	-
DRT	0114	io.cpp:3199	ERROR	-
DRT	0115	FlexPA_graphics.cpp:66	INFO	-
DRT	0116	io.cpp:1153	ERROR	-
DRT	0117	io.cpp:1156	ERROR	-
DRT	0118	TritonRoute.tcl:269	ERROR	-
DRT	0119	FlexPA_graphics.cpp:270	INFO	-
DRT	0122	io.cpp:2251	WARN	-
DRT	0123	io.cpp:2288	WARN	-
DRT	0124	io.cpp:2516	WARN	-
DRT	0125	io.cpp:2540	ERROR	-
DRT	0126	io.cpp:2547	ERROR	-
DRT	0127	io.cpp:2557	ERROR	-
DRT	0128	io.cpp:2399	ERROR	-
DRT	0129	io.cpp:2405	ERROR	-
DRT	0130	io.cpp:2418	ERROR	-
DRT	0131	io.cpp:2439	WARN	-
DRT	0132	io.cpp:2460	WARN	-
DRT	0133	io.cpp:2470	WARN	-
DRT	0134	io.cpp:2485	WARN	-
DRT	0135	io.cpp:2495	WARN	-
DRT	0136	io.cpp:2607	ERROR	-
DRT	0138	io.cpp:1900	WARN	-
DRT	0139	io.cpp:1842	WARN	-
DRT	0140	io.cpp:1872	WARN	-
DRT	0141	io.cpp:1874	WARN	-
DRT	0142	io.cpp:1876	WARN	-
DRT	0143	io.cpp:1878	WARN	-
DRT	0144	io.cpp:1916	WARN	-
DRT	0145	io.cpp:1970	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
DRT	0146	io.cpp:1999	WARN	-
DRT	0147	io.cpp:2126	WARN	-
DRT	0149	io.cpp:2602	INFO	-
DRT	0150	io.cpp:1148	INFO	-
DRT	0153	io.cpp:2665	ERROR	-
DRT	0154	io.cpp:2669	ERROR	-
DRT	0155	io.cpp:2686	ERROR	-
DRT	0156	io.cpp:2709	INFO	-
DRT	0157	io.cpp:2713	INFO	-
DRT	0160	io_parser_helper.cpp:135	WARN	-
DRT	0161	io.cpp:1579	WARN	-
DRT	0162	io_pin.cpp:37	INFO	-
DRT	0163	io_pin.cpp:83	INFO	-
DRT	0164	io_pin.cpp:137	INFO	-
DRT	0165	FlexPA.cpp:177	INFO	-
DRT	0166	FlexPA.cpp:214	INFO	-
DRT	0167	frTechObject.h:259	INFO	-
DRT	0168	io_parser_helper.cpp:792	INFO	-
DRT	0169	io_parser_helper.cpp:804	INFO	-
DRT	0170	io_parser_helper.cpp:989	ERROR	-
DRT	0171	io_parser_helper.cpp:994	ERROR	-
DRT	0172	io_parser_helper.cpp:1048	ERROR	-
DRT	0173	io_parser_helper.cpp:1053	ERROR	-
DRT	0174	io_parser_helper.cpp:1093	ERROR	-
DRT	0175	io_parser_helper.cpp:1109	ERROR	-
DRT	0176	io_parser_helper.cpp:1115	INFO	-
DRT	0177	io_parser_helper.cpp:1121	INFO	-
DRT	0178	io_parser_helper.cpp:842	INFO	-
DRT	0179	io_parser_helper.cpp:845	INFO	-
DRT	0180	io.cpp:3043	INFO	-
DRT	0181	FlexTA.cpp:331	INFO	-
DRT	0182	FlexTA.cpp:344	INFO	-
DRT	0183	FlexTA.cpp:221	INFO	-
DRT	0184	FlexTA.cpp:239	INFO	-
DRT	0185	io_parser_helper.cpp:861	INFO	-
DRT	0186	FlexTA.cpp:305	INFO	-
DRT	0187	FlexDR.cpp:452	INFO	-
DRT	0190	io.cpp:233	WARN	-
DRT	0191	io.cpp:253	WARN	-
DRT	0192	io.cpp:2525	WARN	-
DRT	0194	FlexDR.cpp:466	INFO	-
DRT	0195	FlexDR.cpp:546	INFO	-
DRT	0198	FlexDR.cpp:817	INFO	-
DRT	0199	FlexDR.cpp:758	INFO	-
DRT	0201	FlexGR.cpp:640	ERROR	-
DRT	0202	FlexGR.cpp:668	WARN	-
DRT	0203	FlexGR.cpp:647	WARN	-
DRT	0204	TritonRoute.cpp:1079	WARN	-
DRT	0206	FlexDR_conn.cpp:1308	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
DRT	0207	FlexDR_graphics.cpp:748	INFO	-
DRT	0210	io.cpp:1757	WARN	-
DRT	0214	io_guide.cpp:878	WARN	-
DRT	0215	io_guide.cpp:887	WARN	-
DRT	0216	io_guide.cpp:896	WARN	-
DRT	0217	io_guide.cpp:901	WARN	-
DRT	0218	io_guide.cpp:938	ERROR	-
DRT	0219	io_guide.cpp:941	ERROR	-
DRT	0220	io_guide.cpp:986	WARN	-
DRT	0221	io_guide.cpp:1005	WARN	-
DRT	0222	io_guide.cpp:1013	WARN	-
DRT	0223	io_guide.cpp:1062	ERROR	-
DRT	0224	io_guide.cpp:1267	WARN	-
DRT	0225	io_guide.cpp:1277	WARN	-
DRT	0226	FlexGC_eol.cpp:477	ERROR	-
DRT	0228	io_guide.cpp:367	ERROR	-
DRT	0229	io_guide.cpp:495	ERROR	-
DRT	0230	io_guide.cpp:582	WARN	-
DRT	0231	io_guide.cpp:598	WARN	-
DRT	0232	io_guide.cpp:610	ERROR	-
DRT	0233	io_parser_helper.cpp:101	ERROR	-
DRT	0234	io_parser_helper.cpp:94	ERROR	-
DRT	0235	io_parser_helper.cpp:206	WARN	-
DRT	0236	io_parser_helper.cpp:213	INFO	-
DRT	0237	io_parser_helper.cpp:231	WARN	-
DRT	0238	io_parser_helper.cpp:238	INFO	-
DRT	0239	io_parser_helper.cpp:295	ERROR	-
DRT	0240	io_parser_helper.cpp:302	WARN	-
DRT	0241	io_parser_helper.cpp:309	WARN	-
DRT	0242	io_parser_helper.cpp:317	ERROR	-
DRT	0243	io_parser_helper.cpp:329	ERROR	-
DRT	0244	io_parser_helper.cpp:334	WARN	-
DRT	0245	io_parser_helper.cpp:850	INFO	-
DRT	0246	io_parser_helper.cpp:894	WARN	-
DRT	0247	io.cpp:2741	WARN	-
DRT	0248	io_pin.cpp:72	WARN	-
DRT	0249	FlexDR_graphics.cpp:669	INFO	-
DRT	0250	FlexDR_graphics.cpp:672	INFO	-
DRT	0253	FlexGC_init.cpp:179	ERROR	-
DRT	0255	FlexDR_maze.cpp:1749	ERROR	-
DRT	0256	io.cpp:375	WARN	-
DRT	0258	io.cpp:1572	WARN	-
DRT	0259	io.cpp:1586	WARN	-
DRT	0261	io.cpp:1607	WARN	-
DRT	0262	io.cpp:1614	WARN	-
DRT	0263	io.cpp:1621	WARN	-
DRT	0267	frTime.cpp:48	INFO	-
DRT	0268	FlexTA.cpp:287	INFO	-
DRT	0269	FlexGC_eol.cpp:58	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
DRT	0270	FlexGC_eol.cpp:232	ERROR	-
DRT	0271	FlexGC_eol.cpp:402	ERROR	-
DRT	0272	io.cpp:2619	WARN	-
DRT	0273	io.cpp:2631	WARN	-
DRT	0275	FlexDR_graphics.cpp:675	INFO	-
DRT	0276	FlexPA_prep.cpp:2083	ERROR	-
DRT	0277	FlexPA_prep.cpp:2850	ERROR	-
DRT	0278	FlexPA_prep.cpp:2888	ERROR	-
DRT	0279	io.cpp:1632	WARN	-
DRT	0280	FlexPA_graphics.cpp:343	WARN	-
DRT	0281	FlexPA_graphics.cpp:349	INFO	-
DRT	0282	io.cpp:176	WARN	-
DRT	0290	TritonRoute.cpp:1284	WARN	-
DRT	0291	TritonRoute.cpp:1351	ERROR	-
DRT	0292	FlexPA_graphics.cpp:305	INFO	-
DRT	0293	FlexPA_graphics.cpp:74	ERROR	-
DRT	0294	io.cpp:3279	ERROR	-
DRT	0295	io.cpp:3283	ERROR	-
DRT	0296	io.cpp:3286	ERROR	-
DRT	0297	io.cpp:3312	ERROR	-
DRT	0298	io.cpp:3318	ERROR	-
DRT	0299	io.cpp:3321	ERROR	-
DRT	0300	io.cpp:3333	ERROR	-
DRT	0301	io.cpp:3344	ERROR	-
DRT	0302	io.cpp:960	ERROR	-
DRT	0303	io.cpp:3351	ERROR	-
DRT	0304	TritonRoute.cpp:829	ERROR	-
DRT	0305	io.cpp:524	ERROR	-
DRT	0306	io.cpp:544	ERROR	-
DRT	0307	io.cpp:568	ERROR	-
DRT	0308	TritonRoute.tcl:461	ERROR	-
DRT	0311	FlexGC_main.cpp:1717	WARN	-
DRT	0312	FlexGC_main.cpp:1724	WARN	-
DRT	0313	FlexGC_main.cpp:1729	WARN	-
DRT	0314	FlexGC_main.cpp:1732	WARN	-
DRT	0315	FlexGC_main.cpp:1735	WARN	-
DRT	0316	FlexGC_main.cpp:1738	WARN	-
DRT	0317	io.cpp:2025	WARN	-
DRT	0318	io.cpp:2033	WARN	-
DRT	0319	io.cpp:2041	WARN	-
DRT	0320	FlexPA_unique.cpp:222	ERROR	-
DRT	0321	FlexPA_unique.cpp:232	ERROR	-
DRT	0322	FlexPA_unique.cpp:240	ERROR	-
DRT	0323	io.cpp:2237	WARN	-
DRT	0324	io.cpp:1672	WARN	-
DRT	0325	io.cpp:1680	WARN	-
DRT	0326	io.cpp:1688	WARN	-
DRT	0327	io.cpp:1696	WARN	-
DRT	0328	io.cpp:1704	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
DRT	0329	FlexPA_prep.cpp:1639	ERROR	-
DRT	0330	FlexPA_prep.cpp:1804	ERROR	-
DRT	0331	FlexPA.cpp:150	ERROR	-
DRT	0332	FlexPA_prep.cpp:1684	ERROR	-
DRT	0400	io.cpp:1296	WARN	-
DRT	0401	io.cpp:1304	WARN	-
DRT	0403	io.cpp:1312	WARN	-
DRT	0404	io.cpp:1084	ERROR	-
DRT	0405	io.cpp:1087	ERROR	-
DRT	0406	FlexTA_assign.cpp:613	ERROR	-
DRT	0410	FlexGC_main.cpp:1855	ERROR	-
DRT	0411	FlexGC_main.cpp:1860	ERROR	-
DRT	0412	FlexTA_assign.cpp:750	ERROR	-
DRT	0415	TritonRoute.cpp:1113	ERROR	-
DRT	0416	io_parser_helper.cpp:609	ERROR	-
DRT	0417	io_parser_helper.cpp:661	ERROR	-
DRT	0418	io_parser_helper.cpp:767	WARN	-
DRT	0419	io_parser_helper.cpp:770	WARN	-
DRT	0421	io_parser_helper.cpp:729	WARN	-
DRT	0422	io_parser_helper.cpp:732	WARN	-
DRT	0500	FlexDR.cpp:1255	ERROR	-
DRT	0506	TritonRoute.tcl:168	ERROR	-
DRT	0507	TritonRoute.tcl:173	ERROR	-
DRT	0508	TritonRoute.tcl:178	ERROR	-
DRT	0512	frRegionQuery.cpp:300	ERROR	-
DRT	0513	frRegionQuery.cpp:229	ERROR	-
DRT	0516	TritonRoute.tcl:183	ERROR	-
DRT	0517	TritonRoute.tcl:371	ERROR	-
DRT	0519	FlexPA_prep.cpp:944	WARN	-
DRT	0520	TritonRoute.tcl:377	ERROR	-
DRT	0550	FlexGridGraph.h:1062	ERROR	-
DRT	0551	FlexGridGraph.h:1074	ERROR	-
DRT	0552	TritonRoute.tcl:335	ERROR	-
DRT	0553	TritonRoute.tcl:340	ERROR	-
DRT	0554	TritonRoute.tcl:345	ERROR	-
DRT	0555	TritonRoute.tcl:350	ERROR	-
DRT	0606	TritonRoute.cpp:550	WARN	-
DRT	0607	TritonRoute.cpp:562	WARN	-
DRT	0608	io_parser_helper.cpp:57	ERROR	-
DRT	0610	TritonRoute.cpp:1181	ERROR	-
DRT	0611	TritonRoute.cpp:1187	ERROR	-
DRT	0612	TritonRoute.tcl:480	ERROR	-
DRT	0613	TritonRoute.tcl:487	ERROR	-
DRT	0615	TritonRoute.cpp:1196	ERROR	-
DRT	0616	TritonRoute.cpp:1201	ERROR	-
DRT	0617	TritonRoute.cpp:574	WARN	-
DRT	0618	TritonRoute.cpp:1204	ERROR	-
DRT	0999	FlexDR.cpp:1273	ERROR	-
DRT	1000	io_guide.cpp:103	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
DRT	1001	io_guide.cpp:151	WARN	-
DRT	1002	io_guide.cpp:214	ERROR	-
DRT	1003	FlexPA_prep.cpp:403	ERROR	-
DRT	1004	FlexPA_prep.cpp:410	ERROR	-
DRT	1005	FlexPA_prep.cpp:421	ERROR	-
DRT	1006	FlexDR_maze.cpp:1801	ERROR	-
DRT	1007	io_guide.cpp:101	ERROR	-
DRT	1008	io_guide.cpp:271	ERROR	-
DRT	1009	FlexDR_init.cpp:1289	ERROR	-
DRT	1010	FlexDR_init.cpp:62	ERROR	-
DRT	1011	io.cpp:1123	ERROR	-
DRT	12304	TritonRoute.cpp:756	ERROR	-
DRT	2000	FlexDR_maze.cpp:2076	INFO	-
DRT	2001	FlexDR_maze.cpp:1531	INFO	-
DRT	2002	FlexDR_maze.cpp:1729	INFO	-
DRT	2003	FlexDR_maze.cpp:1791	INFO	-
DRT	2005	FlexDR_maze.cpp:1964	INFO	-
DRT	2006	FlexDR_maze.cpp:1967	INFO	-
DRT	2007	FlexDR_maze.cpp:1998	INFO	-
DRT	2008	TritonRoute.tcl:257	ERROR	-
DRT	3000	io_guide.cpp:291	ERROR	-
DRT	4000	FlexPA_graphics.cpp:79	INFO	-
DRT	4500	FlexGC_main.cpp:1832	ERROR	-
DRT	4501	FlexGC_main.cpp:1844	ERROR	-
DRT	5000	FlexPA_graphics.cpp:86	WARN	-
DRT	6000	FlexPA_prep.cpp:773	WARN	-
DRT	6001	FlexDR_conn.cpp:1016	WARN	-
DRT	7461	FlexDR.cpp:1231	ERROR	-
DRT	9199	TritonRoute.cpp:474	ERROR	-
DRT	9504	TritonRoute.cpp:788	ERROR	-
DRT	9999	TritonRoute.cpp:413	ERROR	-
DST	0001	Distributed.cc:92	ERROR	-
DST	0002	Distributed.tcl:42	ERROR	-
DST	0003	Distributed.tcl:47	ERROR	-
DST	0004	WorkerConnection.cc:122	WARN	-
DST	0005	WorkerConnection.cc:112	WARN	-
DST	0006	BalancerConnection.cc:102	WARN	-
DST	0007	LoadBalancer.cc:43	INFO	-
DST	0008	BalancerConnection.cc:232	WARN	-
DST	0009	Distributed.cc:110	ERROR	-
DST	0010	Distributed.tcl:66	ERROR	-
DST	0011	Distributed.tcl:71	ERROR	-
DST	0012	Distributed.cc:226	WARN	-
DST	0013	Distributed.cc:236	WARN	-
DST	0014	Distributed.cc:270	WARN	-
DST	0016	Distributed.tcl:91	ERROR	-
DST	0017	Distributed.tcl:96	ERROR	-
DST	0020	Distributed.cc:279	WARN	-
DST	0022	Distributed.cc:289	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
DST	0041	WorkerConnection.cc:78	WARN	-
DST	0042	BalancerConnection.cc:87	WARN	-
DST	0112	Distributed.cc:165	WARN	-
DST	0113	Distributed.cc:175	WARN	-
DST	0114	Distributed.cc:201	WARN	-
DST	0203	LoadBalancer.cc:199	WARN	-
DST	0204	BalancerConnection.cc:137	WARN	-
DST	0205	BalancerConnection.cc:147	WARN	-
DST	0207	BalancerConnection.cc:210	WARN	-
DST	9999	Distributed.cc:256	ERROR	-
FIN	0001	DensityFill.cpp:179	ERROR	-
FIN	0002	DensityFill.cpp:190	ERROR	-
FIN	0003	DensityFill.cpp:450	INFO	-
FIN	0004	DensityFill.cpp:488	INFO	-
FIN	0005	DensityFill.cpp:507	INFO	-
FIN	0006	DensityFill.cpp:513	INFO	-
FIN	0007	finale.tcl:49	ERROR	-
FIN	0008	finale.tcl:55	ERROR	-
FIN	0009	DensityFill.cpp:475	INFO	-
FIN	0010	DensityFill.cpp:533	WARN	-
GPL	0002	placerBase.cpp:816	INFO	-
GPL	0003	placerBase.cpp:843	INFO	-
GPL	0004	placerBase.cpp:844	INFO	-
GPL	0005	placerBase.cpp:845	INFO	-
GPL	0006	placerBase.cpp:1310	INFO	-
GPL	0007	placerBase.cpp:1314	INFO	-
GPL	0008	placerBase.cpp:1315	INFO	-
GPL	0009	placerBase.cpp:1316	INFO	-
GPL	0010	placerBase.cpp:1317	INFO	-
GPL	0011	placerBase.cpp:1318	INFO	-
GPL	0012	placerBase.cpp:1320	INFO	-
GPL	0013	placerBase.cpp:1321	INFO	-
GPL	0014	placerBase.cpp:1322	INFO	-
GPL	0015	placerBase.cpp:1323	INFO	-
GPL	0016	placerBase.cpp:1329	INFO	-
GPL	0017	placerBase.cpp:1330	INFO	-
GPL	0018	placerBase.cpp:1332	INFO	-
GPL	0019	placerBase.cpp:1333	INFO	-
GPL	0020	placerBase.cpp:1335	INFO	-
GPL	0021	placerBase.cpp:1336	INFO	-
GPL	0023	nesterovBase.cpp:754	INFO	-
GPL	0024	nesterovBase.cpp:755	INFO	-
GPL	0025	nesterovBase.cpp:756	INFO	-
GPL	0026	nesterovBase.cpp:757	INFO	-
GPL	0027	nesterovBase.cpp:758	INFO	-
GPL	0028	nesterovBase.cpp:788	INFO	-
GPL	0029	nesterovBase.cpp:793	INFO	-
GPL	0030	nesterovBase.cpp:809	INFO	-
GPL	0031	nesterovBase.cpp:1567	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
GPL	0032	nesterovBase.cpp:1568	INFO	-
GPL	0033	nesterovBase.cpp:1569	INFO	-
GPL	0034	nesterovBase.cpp:1915	INFO	-
GPL	0035	nesterovBase.cpp:1933	INFO	-
GPL	0036	routeBase.cpp:198	INFO	-
GPL	0037	routeBase.cpp:199	INFO	-
GPL	0038	routeBase.cpp:200	INFO	-
GPL	0039	routeBase.cpp:201	INFO	-
GPL	0040	routeBase.cpp:223	INFO	-
GPL	0045	routeBase.cpp:665	INFO	-
GPL	0046	routeBase.cpp:666	INFO	-
GPL	0047	routeBase.cpp:684	INFO	-
GPL	0048	routeBase.cpp:685	INFO	-
GPL	0049	routeBase.cpp:697	INFO	-
GPL	0050	routeBase.cpp:698	INFO	-
GPL	0051	routeBase.cpp:699	INFO	-
GPL	0052	routeBase.cpp:700	INFO	-
GPL	0053	routeBase.cpp:704	INFO	-
GPL	0054	routeBase.cpp:723	INFO	-
GPL	0055	routeBase.cpp:724	INFO	-
GPL	0056	routeBase.cpp:725	INFO	-
GPL	0057	routeBase.cpp:726	INFO	-
GPL	0058	routeBase.cpp:728	INFO	-
GPL	0059	routeBase.cpp:729	INFO	-
GPL	0063	routeBase.cpp:795	INFO	-
GPL	0064	routeBase.cpp:796	INFO	-
GPL	0065	routeBase.cpp:797	INFO	-
GPL	0066	routeBase.cpp:852	INFO	-
GPL	0067	routeBase.cpp:853	INFO	-
GPL	0068	routeBase.cpp:854	INFO	-
GPL	0069	routeBase.cpp:855	INFO	-
GPL	0070	routeBase.cpp:857	INFO	-
GPL	0071	routeBase.cpp:858	INFO	-
GPL	0072	routeBase.cpp:859	INFO	-
GPL	0073	routeBase.cpp:860	INFO	-
GPL	0074	routeBase.cpp:868	INFO	-
GPL	0075	routeBase.cpp:881	INFO	-
GPL	0100	timingBase.cpp:166	INFO	-
GPL	0102	timingBase.cpp:169	WARN	-
GPL	0103	timingBase.cpp:207	INFO	-
GPL	0114	timingBase.cpp:156	WARN	-
GPL	0115	replace.tcl:146	WARN	-
GPL	0116	replace.tcl:158	WARN	-
GPL	0118	placerBase.cpp:835	ERROR	-
GPL	0119	placerBase.cpp:877	ERROR	-
GPL	0120	placerBase.cpp:880	ERROR	-
GPL	0121	replace.tcl:118	ERROR	-
GPL	0122	mbff.cpp:334	ERROR	-
GPL	0130	replace.tcl:319	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
GPL	0131	replace.tcl:414	ERROR	-
GPL	0132	replace.cpp:239	INFO	-
GPL	0133	replace.cpp:256	INFO	-
GPL	0134	placerBase.cpp:106	WARN	-
GPL	0135	replace.tcl:180	ERROR	-
GPL	0136	replace.cpp:339	WARN	-
GPL	0150	replace.tcl:122	WARN	-
GPL	0151	replace.tcl:153	WARN	-
GPL	0250	initialPlace.cpp:120	WARN	-
GPL	0251	initialPlace.cpp:125	WARN	-
GPL	0301	placerBase.cpp:1339	ERROR	-
GPL	0302	nesterovBase.cpp:1670	ERROR	-
GPL	0303	nesterovBase.cpp:1897	ERROR	-
GPL	0304	nesterovPlace.cpp:287	ERROR	-
GPL	0305	placerBase.cpp:829	ERROR	-
GPL	1001	mbff.cpp:1689	INFO	-
GPL	1002	mbff.cpp:1511	INFO	-
GPL	1010	mbff.cpp:1695	INFO	-
GRT	0001	GlobalRouter.cpp:798	INFO	-
GRT	0002	GlobalRouter.cpp:799	INFO	-
GRT	0003	GlobalRouter.cpp:3320	INFO	-
GRT	0004	GlobalRouter.cpp:3039	INFO	-
GRT	0005	GlobalRouter.tcl:504	ERROR	-
GRT	0006	GlobalRouter.cpp:358	INFO	-
GRT	0009	GlobalRouter.cpp:3942	INFO	-
GRT	0012	RepairAntennas.cpp:94	INFO	-
GRT	0014	GlobalRouter.cpp:317	INFO	-
GRT	0015	GlobalRouter.cpp:364	INFO	-
GRT	0018	GlobalRouter.cpp:2101	INFO	-
GRT	0019	GlobalRouter.cpp:2835	INFO	-
GRT	0020	GlobalRouter.cpp:3532	INFO	-
GRT	0021	GlobalRouter.cpp:3533	INFO	-
GRT	0022	GlobalRouter.cpp:3534	INFO	-
GRT	0023	GlobalRouter.cpp:3535	INFO	-
GRT	0025	MakeWireParasitics.cpp:260	WARN	-
GRT	0026	MakeWireParasitics.cpp:366	WARN	-
GRT	0027	RepairAntennas.cpp:353	WARN	-
GRT	0028	GlobalRouter.cpp:3315	ERROR	-
GRT	0029	GlobalRouter.cpp:2931	ERROR	-
GRT	0030	GlobalRouter.cpp:1338	WARN	-
GRT	0031	GlobalRouter.cpp:2055	WARN	-
GRT	0033	GlobalRouter.cpp:2325	WARN	-
GRT	0034	GlobalRouter.cpp:2884	WARN	-
GRT	0035	GlobalRouter.cpp:2914	WARN	-
GRT	0036	GlobalRouter.cpp:2974	WARN	-
GRT	0037	GlobalRouter.cpp:3108	WARN	-
GRT	0038	GlobalRouter.cpp:3263	WARN	-
GRT	0039	GlobalRouter.cpp:3299	WARN	-
GRT	0040	GlobalRouter.cpp:3355	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
GRT	0041	GlobalRouter.cpp:3387	WARN	-
GRT	0042	GlobalRouter.cpp:2999	ERROR	-
GRT	0043	GlobalRouter.cpp:3465	INFO	-
GRT	0044	GlobalRouter.tcl:61	ERROR	-
GRT	0045	GlobalRouter.tcl:339	ERROR	-
GRT	0047	GlobalRouter.tcl:75	ERROR	-
GRT	0048	GlobalRouter.tcl:83	ERROR	-
GRT	0049	GlobalRouter.tcl:89	ERROR	-
GRT	0050	GlobalRouter.tcl:112	ERROR	-
GRT	0051	GlobalRouter.tcl:215	ERROR	-
GRT	0052	GlobalRouter.tcl:219	ERROR	-
GRT	0053	GlobalRouter.cpp:3547	INFO	-
GRT	0054	GlobalRouter.cpp:369	INFO	-
GRT	0055	GlobalRouter.tcl:230	ERROR	-
GRT	0056	GlobalRouter.tcl:577	ERROR	-
GRT	0057	GlobalRouter.tcl:563	ERROR	-
GRT	0059	GlobalRouter.tcl:478	ERROR	-
GRT	0060	GlobalRouter.tcl:490	ERROR	-
GRT	0061	GlobalRouter.tcl:493	ERROR	-
GRT	0062	GlobalRouter.tcl:519	ERROR	-
GRT	0063	GlobalRouter.tcl:526	ERROR	-
GRT	0064	GlobalRouter.tcl:532	ERROR	-
GRT	0065	GlobalRouter.tcl:536	ERROR	-
GRT	0066	GlobalRouter.tcl:540	ERROR	-
GRT	0067	GlobalRouter.tcl:544	ERROR	-
GRT	0068	RepairAntennas.cpp:125	ERROR	-
GRT	0069	GlobalRouter.tcl:304	ERROR	-
GRT	0071	GlobalRouter.cpp:993	ERROR	-
GRT	0072	GlobalRouter.cpp:1201	ERROR	-
GRT	0073	GlobalRouter.tcl:317	ERROR	-
GRT	0074	GlobalRouter.cpp:1819	ERROR	-
GRT	0075	GlobalRouter.cpp:1797	ERROR	-
GRT	0076	GlobalRouter.cpp:1875	ERROR	-
GRT	0078	GlobalRouter.cpp:1940	ERROR	-
GRT	0079	GlobalRouter.cpp:2042	ERROR	-
GRT	0080	GlobalRouter.cpp:2071	ERROR	-
GRT	0082	GlobalRouter.cpp:2459	ERROR	-
GRT	0084	GlobalRouter.cpp:492	ERROR	-
GRT	0085	GlobalRouter.cpp:487	ERROR	-
GRT	0086	GlobalRouter.cpp:2618	ERROR	-
GRT	0088	GlobalRouter.cpp:2663	INFO	-
GRT	0090	GlobalRouter.cpp:2723	ERROR	-
GRT	0094	GlobalRouter.cpp:3329	ERROR	-
GRT	0096	GlobalRouter.cpp:3599	INFO	-
GRT	0101	FastRoute.cpp:994	INFO	-
GRT	0103	FastRoute.cpp:1129	INFO	-
GRT	0111	FastRoute.cpp:1303	INFO	-
GRT	0112	FastRoute.cpp:1304	INFO	-
GRT	0113	FastRoute.cpp:381	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
GRT	0114	FastRoute.cpp:411	WARN	-
GRT	0115	GlobalRouter.cpp:309	WARN	-
GRT	0118	GlobalRouter.cpp:300	ERROR	Do refer to the GUI guide and global routing debugging tips.
GRT	0119	GlobalRouter.cpp:293	ERROR	Do refer to the GUI guide and global routing debugging tips.
GRT	0122	RipUp.cpp:450	ERROR	-
GRT	0123	RipUp.cpp:572	ERROR	-
GRT	0124	GlobalRouter.cpp:2630	ERROR	-
GRT	0125	maze.cpp:830	ERROR	-
GRT	0126	GlobalRouter.cpp:511	ERROR	-
GRT	0146	GlobalRouter.tcl:270	WARN	-
GRT	0147	GlobalRouter.cpp:2643	ERROR	-
GRT	0148	GlobalRouter.cpp:2650	ERROR	-
GRT	0149	utility.cpp:1988	ERROR	-
GRT	0150	maze.cpp:1856	ERROR	-
GRT	0151	maze.cpp:1890	WARN	-
GRT	0152	maze.cpp:2007	WARN	-
GRT	0153	maze.cpp:2042	WARN	-
GRT	0164	utility.cpp:1692	WARN	-
GRT	0165	utility.cpp:1706	WARN	-
GRT	0166	utility.cpp:1722	WARN	-
GRT	0167	utility.cpp:1736	WARN	-
GRT	0169	maze.cpp:1040	ERROR	-
GRT	0170	maze.cpp:1218	WARN	-
GRT	0171	maze3D.cpp:491	ERROR	-
GRT	0172	maze3D.cpp:809	ERROR	-
GRT	0179	route.cpp:338	WARN	-
GRT	0181	route.cpp:1160	WARN	-
GRT	0183	maze3D.cpp:1199	ERROR	-
GRT	0184	maze3D.cpp:787	WARN	-
GRT	0187	maze3D.cpp:477	ERROR	-
GRT	0188	RSMT.cpp:140	ERROR	-
GRT	0189	RSMT.cpp:143	ERROR	-
GRT	0197	utility.cpp:313	INFO	-
GRT	0198	utility.cpp:314	INFO	-
GRT	0199	utility.cpp:315	INFO	-
GRT	0200	utility.cpp:498	WARN	-
GRT	0201	maze.cpp:910	ERROR	-
GRT	0202	utility.cpp:609	ERROR	-
GRT	0203	utility.cpp:1036	ERROR	-
GRT	0204	utility.cpp:1119	ERROR	-
GRT	0206	utility.cpp:1261	ERROR	-
GRT	0207	utility.cpp:1673	WARN	-
GRT	0208	utility.cpp:1681	WARN	-
GRT	0209	GlobalRouter.cpp:2979	ERROR	-
GRT	0214	FastRoute.cpp:587	ERROR	-
GRT	0215	GlobalRouter.tcl:333	WARN	-
GRT	0219	GlobalRouter.tcl:143	ERROR	-
GRT	0220	GlobalRouter.tcl:155	ERROR	-
GRT	0221	RepairAntennas.cpp:197	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
GRT	0222	GlobalRouter.tcl:499	ERROR	-
GRT	0223	GlobalRouter.tcl:381	ERROR	-
GRT	0224	GlobalRouter.tcl:442	ERROR	-
GRT	0225	RipUp.cpp:176	ERROR	-
GRT	0226	RipUp.cpp:267	ERROR	-
GRT	0228	utility.cpp:1569	ERROR	-
GRT	0229	utility.cpp:1583	ERROR	-
GRT	0230	FastRoute.cpp:1274	WARN	-
GRT	0231	GlobalRouter.tcl:424	ERROR	-
GRT	0232	GlobalRouter.cpp:3994	ERROR	-
GRT	0233	GlobalRouter.cpp:1493	ERROR	-
GRT	0234	GlobalRouter.cpp:1516	ERROR	-
GRT	0235	GlobalRouter.cpp:1536	ERROR	-
GRT	0236	GlobalRouter.cpp:1544	ERROR	-
GRT	0237	GlobalRouter.cpp:3737	INFO	-
GRT	0238	GlobalRouter.tcl:470	ERROR	-
GRT	0239	GlobalRouter.cpp:3761	WARN	-
GRT	0240	GlobalRouter.cpp:3767	INFO	-
GRT	0241	GlobalRouter.cpp:3732	WARN	-
GRT	0242	GlobalRouter.tcl:175	ERROR	-
GRT	0243	RepairAntennas.cpp:390	WARN	-
GRT	0244	GlobalRouter.cpp:348	ERROR	-
GRT	0245	GlobalRouter.tcl:320	ERROR	-
GRT	0246	GlobalRouter.cpp:342	WARN	-
GRT	0247	utility.cpp:1539	ERROR	-
GRT	0248	utility.cpp:1550	ERROR	-
GRT	0249	GlobalRouter.cpp:1480	ERROR	-
GRT	0250	GlobalRouter.cpp:1525	WARN	-
GRT	0251	GlobalRouter.cpp:252	ERROR	-
GRT	0252	GlobalRouter.tcl:353	ERROR	-
GRT	0253	GlobalRouter.cpp:1683	ERROR	-
GRT	0300	GlobalRouter.cpp:189	WARN	-
GRT	0301	GlobalRouter.cpp:1322	WARN	-
GRT	0350	MakeWireParasitics.cpp:474	WARN	-
GRT	0500	RipUp.cpp:337	ERROR	-
GRT	0600	utility.cpp:2493	ERROR	-
GRT	1247	utility.cpp:1429	ERROR	-
GRT	1248	utility.cpp:1442	ERROR	-
GUI	0001	gui.i:47	INFO	-
GUI	0002	gui.i:53	ERROR	-
GUI	0003	gui.i:64	ERROR	-
GUI	0005	gui.i:68	ERROR	-
GUI	0006	gui.i:72	ERROR	-
GUI	0007	gui.i:337	ERROR	-
GUI	0008	gui.cpp:1248	WARN	-
GUI	0009	gui.i:360	ERROR	-
GUI	0010	gui_utils.cpp:71	WARN	-
GUI	0011	gui_utils.cpp:121	WARN	-
GUI	0012	gui_utils.cpp:126	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
GUI	0013	displayControls.cpp:1064	ERROR	-
GUI	0014	displayControls.cpp:1085	WARN	-
GUI	0015	gui.cpp:682	ERROR	-
GUI	0016	gui.cpp:715	ERROR	-
GUI	0017	gui.tcl:120	ERROR	-
GUI	0018	gui.tcl:133	ERROR	-
GUI	0019	gui.tcl:342	ERROR	-
GUI	0020	gui.tcl:48	ERROR	-
GUI	0021	gui.tcl:53	ERROR	-
GUI	0022	mainWindow.cpp:725	ERROR	-
GUI	0023	mainWindow.cpp:1209	WARN	-
GUI	0024	mainWindow.cpp:1030	WARN	-
GUI	0025	mainWindow.cpp:816	ERROR	-
GUI	0026	gui.tcl:79	ERROR	-
GUI	0027	gui.tcl:84	ERROR	-
GUI	0028	gui.cpp:878	ERROR	-
GUI	0029	gui.cpp:907	ERROR	-
GUI	0030	drcWidget.cpp:517	ERROR	-
GUI	0031	gui.tcl:125	WARN	-
GUI	0032	drcWidget.cpp:500	ERROR	-
GUI	0033	gui.cpp:290	WARN	-
GUI	0034	displayControls.cpp:1097	WARN	-
GUI	0035	gui.cpp:503	ERROR	-
GUI	0036	gui.i:497	ERROR	-
GUI	0037	gui.i:503	ERROR	-
GUI	0038	gui.tcl:211	ERROR	-
GUI	0039	gui.tcl:239	WARN	-
GUI	0040	drcWidget.cpp:594	WARN	-
GUI	0041	drcWidget.cpp:688	WARN	-
GUI	0042	drcWidget.cpp:677	WARN	-
GUI	0043	drcWidget.cpp:666	WARN	-
GUI	0044	drcWidget.cpp:655	WARN	-
GUI	0045	drcWidget.cpp:548	ERROR	-
GUI	0046	drcWidget.cpp:563	ERROR	-
GUI	0047	drcWidget.cpp:583	ERROR	-
GUI	0048	drcWidget.cpp:613	ERROR	-
GUI	0049	drcWidget.cpp:619	ERROR	-
GUI	0050	drcWidget.cpp:626	ERROR	-
GUI	0051	drcWidget.cpp:715	WARN	-
GUI	0052	drcWidget.cpp:709	WARN	-
GUI	0053	gui.cpp:1151	ERROR	-
GUI	0054	displayControls.cpp:787	WARN	-
GUI	0055	drcWidget.cpp:755	ERROR	-
GUI	0056	gui.tcl:230	ERROR	-
GUI	0057	displayControls.cpp:709	WARN	-
GUI	0058	drcWidget.cpp:818	ERROR	-
GUI	0059	gui.cpp:483	ERROR	-
GUI	0060	gui.cpp:926	ERROR	-
GUI	0061	gui.cpp:935	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
GUI	0062	gui.cpp:944	ERROR	-
GUI	0063	gui.cpp:951	ERROR	-
GUI	0064	gui.cpp:695	ERROR	-
GUI	0065	gui.cpp:700	ERROR	-
GUI	0066	heatMap.cpp:579	WARN	-
GUI	0067	gui.tcl:273	ERROR	-
GUI	0068	gui.tcl:282	ERROR	-
GUI	0069	gui.tcl:285	ERROR	-
GUI	0070	gui.tcl:316	ERROR	-
GUI	0071	gui.tcl:323	ERROR	-
GUI	0072	heatMap.cpp:94	ERROR	-
GUI	0073	heatMap.cpp:99	ERROR	-
GUI	0074	clockWidget.cpp:1478	ERROR	-
GUI	0075	gui.cpp:486	ERROR	-
GUI	0076	mainWindow.cpp:1600	ERROR	-
GUI	0077	mainWindow.cpp:1594	WARN	-
GUI	0078	layoutViewer.cpp:1990	WARN	-
GUI	0079	drcWidget.cpp:791	WARN	-
GUI	0080	drcWidget.cpp:876	WARN	-
GUI	0081	drcWidget.cpp:869	WARN	-
GUI	0082	drcWidget.cpp:856	WARN	-
GUI	0083	drcWidget.cpp:849	WARN	-
GUI	0084	drcWidget.cpp:841	WARN	-
GUI	0085	drcWidget.cpp:834	WARN	-
GUI	0086	drcWidget.cpp:779	ERROR	-
GUI	0087	drcWidget.cpp:765	ERROR	-
GUI	0088	gui.tcl:189	ERROR	-
GUI	0089	clockWidget.cpp:1443	ERROR	-
GUI	0090	gui.i:591	ERROR	-
GUI	0091	gui.i:604	ERROR	-
GUI	0092	gui.i:617	ERROR	-
GUI	0093	gui.i:630	ERROR	-
GUI	0094	layoutViewer.cpp:1953	WARN	-
GUI	0095	gui.cpp:977	ERROR	-
GUI	0096	gui.tcl:140	ERROR	-
GUI	0098	gui.tcl:145	ERROR	-
GUI	0099	drcWidget.cpp:805	WARN	-
IFP	0001	InitFloorplan.cc:475	INFO	-
IFP	0010	InitFloorplan.tcl:159	ERROR	-
IFP	0011	InitFloorplan.tcl:56	ERROR	-
IFP	0013	InitFloorplan.tcl:80	ERROR	-
IFP	0015	InitFloorplan.tcl:103	ERROR	-
IFP	0016	InitFloorplan.tcl:115	ERROR	-
IFP	0017	InitFloorplan.tcl:131	ERROR	-
IFP	0019	InitFloorplan.tcl:134	ERROR	-
IFP	0021	InitFloorplan.cc:615	WARN	-
IFP	0022	InitFloorplan.cc:623	WARN	-
IFP	0025	InitFloorplan.tcl:162	ERROR	-
IFP	0026	InitFloorplan.cc:321	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
IFP	0027	InitFloorplan.cc:352	WARN	-
IFP	0028	InitFloorplan.cc:216	WARN	-
IFP	0029	InitFloorplan.cc:536	ERROR	-
IFP	0030	InitFloorplan.cc:561	INFO	-
IFP	0031	InitFloorplan.tcl:228	ERROR	-
IFP	0032	InitFloorplan.tcl:233	ERROR	-
IFP	0038	InitFloorplan.i:62	ERROR	-
IFP	0039	InitFloorplan.cc:520	ERROR	-
IFP	0040	InitFloorplan.cc:430	ERROR	-
IFP	0042	InitFloorplan.cc:181	ERROR	-
IFP	0043	InitFloorplan.cc:405	WARN	-
IFP	0044	InitFloorplan.cc:680	ERROR	-
IFP	0045	InitFloorplan.cc:693	ERROR	-
MPL	0001	mpl.tcl:84	ERROR	-
MPL	0002	hier_rtlmp.cpp:988	ERROR	-
MPL	0003	hier_rtlmp.cpp:2792	ERROR	-
MPL	0004	hier_rtlmp.cpp:3016	ERROR	-
MPL	0005	hier_rtlmp.cpp:3821	ERROR	-
MPL	0006	hier_rtlmp.cpp:4070	ERROR	-
MPL	0007	hier_rtlmp.cpp:5205	ERROR	-
MPL	0008	hier_rtlmp.cpp:5224	ERROR	-
MPL	0009	hier_rtlmp.cpp:5371	ERROR	-
MPL	0010	hier_rtlmp.cpp:5570	ERROR	-
MPL	0011	hier_rtlmp.cpp:6377	ERROR	-
MPL	0012	mpl.tcl:285	ERROR	-
MPL	0013	bus_synthesis.cpp:1114	INFO	-
MPL	0014	bus_synthesis.cpp:1128	INFO	-
MPL	0015	graphics.cpp:327	ERROR	-
MPL	0016	hier_rtlmp.cpp:434	ERROR	-
MPL	0017	hier_rtlmp.cpp:429	INFO	-
MPL	0018	mpl.tcl:295	WARN	-
MPL	0019	mpl.tcl:273	ERROR	-
MPL	0020	mpl.tcl:315	ERROR	-
MPL	0021	mpl.tcl:317	ERROR	-
MPL	0022	mpl.tcl:281	ERROR	-
MPL	0024	hier_rtlmp.cpp:496	INFO	-
MPL	0025	hier_rtlmp.cpp:502	WARN	-
MPL	0026	hier_rtlmp.cpp:503	WARN	-
MPL	0027	hier_rtlmp.cpp:629	INFO	-
MPL	0028	hier_rtlmp.cpp:650	INFO	-
MPL	0034	rtl_mp.cpp:143	ERROR	-
MPL	0035	rtl_mp.cpp:186	INFO	-
MPL	0036	rtl_mp.cpp:176	WARN	-
MPL	0037	hier_rtlmp.cpp:696	INFO	-
MPL	0038	object.cpp:626	ERROR	-
MPL	0039	hier_rtlmp.cpp:638	INFO	-
MPL	0040	hier_rtlmp.cpp:5072	ERROR	-
MPL	0061	MacroPlacer.cpp:432	WARN	-
MPL	0064	MacroPlacer.cpp:709	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
MPL	0065	MacroPlacer.cpp:1287	WARN	-
MPL	0066	MacroPlacer.cpp:232	WARN	-
MPL	0067	MacroPlacer.cpp:208	INFO	-
MPL	0068	MacroPlacer.cpp:230	INFO	-
MPL	0069	MacroPlacer.cpp:279	INFO	-
MPL	0070	MacroPlacer.cpp:404	INFO	-
MPL	0071	MacroPlacer.cpp:452	INFO	-
MPL	0072	MacroPlacer.cpp:489	WARN	-
MPL	0073	MacroPlacer.cpp:482	INFO	-
MPL	0076	MacroPlacer.cpp:575	INFO	-
MPL	0077	MacroPlacer.cpp:628	INFO	-
MPL	0079	MacroPlacer.cpp:640	INFO	-
MPL	0080	MacroPlacer.cpp:776	INFO	-
MPL	0085	MacroPlacer.tcl:85	WARN	-
MPL	0089	MacroPlacer.tcl:69	ERROR	-
MPL	0092	MacroPlacer.tcl:49	ERROR	-
MPL	0093	MacroPlacer.tcl:60	ERROR	-
MPL	0094	MacroPlacer.tcl:80	ERROR	-
MPL	0095	MacroPlacer.tcl:102	ERROR	-
MPL	0096	MacroPlacer.tcl:115	ERROR	-
MPL	0097	MacroPlacer.cpp:522	ERROR	-
MPL	0098	MacroPlacer.cpp:164	WARN	-
MPL	0099	MacroPlacer.cpp:808	ERROR	-
MPL	0100	MacroPlacer.cpp:829	WARN	-
MPL	0101	MacroPlacer.cpp:833	INFO	-
MPL	0102	MacroPlacer.cpp:196	INFO	-
ODB	0000	dbWireCodec.cpp:1483	ERROR	-
ODB	0002	dbDatabase.cpp:536	ERROR	-
ODB	0005	dbBlock.cpp:1920	WARN	-
ODB	0006	dbBlock.cpp:1945	WARN	-
ODB	0007	dbBlock.cpp:1970	WARN	-
ODB	0008	dbBlock.cpp:2544	ERROR	-
ODB	0009	dbBlock.cpp:2548	WARN	-
ODB	0010	dbBlock.cpp:2795	INFO	-
ODB	0011	dbBlock.cpp:2936	WARN	-
ODB	0012	dbBlock.cpp:2946	ERROR	-
ODB	0013	dbBlock.cpp:3101	ERROR	-
ODB	0014	dbBlock.cpp:3275	WARN	-
ODB	0015	tmg_conn_w.cpp:98	INFO	-
ODB	0016	tmg_conn.cpp:1667	ERROR	-
ODB	0018	tmg_conn.cpp:1890	ERROR	-
ODB	0019	dbBlock.cpp:3356	WARN	-
ODB	0021	dbCCSeg.cpp:408	INFO	-
ODB	0022	dbCCSeg.cpp:434	INFO	-
ODB	0023	dbCCSeg.cpp:755	ERROR	-
ODB	0024	dbCapNode.cpp:182	WARN	-
ODB	0025	dbCapNode.cpp:1082	INFO	-
ODB	0032	upf.i:139	ERROR	-
ODB	0034	dbITerm.cpp:589	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
ODB	0036	dbInst.cpp:690	INFO	-
ODB	0037	dbInst.cpp:1000	INFO	-
ODB	0038	dbInst.cpp:1021	WARN	-
ODB	0039	dbInst.cpp:1026	WARN	-
ODB	0040	dbInst.cpp:1034	WARN	-
ODB	0041	dbInst.cpp:1039	WARN	-
ODB	0042	dbInst.cpp:1047	WARN	-
ODB	0043	dbInst.cpp:1056	WARN	-
ODB	0044	dbInst.cpp:1167	WARN	-
ODB	0045	dbInst.cpp:1215	WARN	-
ODB	0046	dbInst.cpp:1242	WARN	-
ODB	0047	dbInst.cpp:1593	WARN	-
ODB	0048	dbNet.cpp:1013	WARN	-
ODB	0049	dbNet.cpp:2132	ERROR	-
ODB	0050	dbNet.cpp:2138	ERROR	-
ODB	0051	dbNet.cpp:2162	ERROR	-
ODB	0052	dbNet.cpp:2577	WARN	-
ODB	0053	dbNet.cpp:2623	WARN	-
ODB	0054	dbRSeg.cpp:553	INFO	-
ODB	0056	dbRSeg.cpp:559	INFO	-
ODB	0057	dbRSeg.cpp:836	WARN	-
ODB	0058	dbTech.cpp:974	WARN	-
ODB	0059	dbTech.cpp:983	ERROR	-
ODB	0060	dbTech.cpp:992	ERROR	-
ODB	0061	dbTech.cpp:1001	ERROR	-
ODB	0062	dbWire.cpp:1776	WARN	-
ODB	0063	dbWireCodec.cpp:1224	WARN	-
ODB	0064	dbWireCodec.cpp:1233	INFO	-
ODB	0065	dbWireCodec.cpp:1242	INFO	-
ODB	0066	dbWireCodec.cpp:1254	INFO	-
ODB	0067	dbWireCodec.cpp:1263	INFO	-
ODB	0068	dbWireCodec.cpp:1286	INFO	-
ODB	0069	dbWireCodec.cpp:1298	INFO	-
ODB	0070	dbWireCodec.cpp:1310	INFO	-
ODB	0071	dbWireCodec.cpp:1320	INFO	-
ODB	0072	dbWireCodec.cpp:1329	WARN	-
ODB	0073	dbWireCodec.cpp:1348	INFO	-
ODB	0074	dbWireCodec.cpp:1355	INFO	-
ODB	0075	dbWireCodec.cpp:1370	INFO	-
ODB	0076	dbWireCodec.cpp:1377	INFO	-
ODB	0077	dbWireCodec.cpp:1384	INFO	-
ODB	0078	dbWireCodec.cpp:1390	INFO	-
ODB	0079	dbWireCodec.cpp:1396	INFO	-
ODB	0080	dbWireCodec.cpp:1405	INFO	-
ODB	0081	dbWireCodec.cpp:1419	INFO	-
ODB	0082	dbWireCodec.cpp:1424	INFO	-
ODB	0083	dbWireCodec.cpp:1432	ERROR	-
ODB	0084	dbWireCodec.cpp:1441	INFO	-
ODB	0085	dbWireCodec.cpp:1450	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
ODB	0086	dbWireCodec.cpp:1458	ERROR	-
ODB	0087	dbWirePathItr.cpp:515	WARN	-
ODB	0088	definBlockage.cpp:73	WARN	-
ODB	0089	definBlockage.cpp:84	WARN	-
ODB	0090	definBlockage.cpp:206	WARN	-
ODB	0091	definBlockage.cpp:227	WARN	-
ODB	0092	definComponent.cpp:151	WARN	-
ODB	0093	definComponent.cpp:173	WARN	-
ODB	0094	definComponent.cpp:179	INFO	-
ODB	0095	definFill.cpp:64	WARN	-
ODB	0096	definNet.cpp:122	WARN	-
ODB	0097	definNet.cpp:147	INFO	-
ODB	0098	definNet.cpp:161	WARN	-
ODB	0099	definNet.cpp:200	WARN	-
ODB	0100	definNet.cpp:210	WARN	-
ODB	0102	definNet.cpp:264	WARN	-
ODB	0103	definNet.cpp:278	WARN	-
ODB	0104	definNet.cpp:363	WARN	-
ODB	0105	definNet.cpp:378	WARN	-
ODB	0106	definNet.cpp:404	WARN	-
ODB	0107	definNet.cpp:518	WARN	-
ODB	0108	definNet.cpp:550	WARN	-
ODB	0109	definNet.cpp:574	WARN	-
ODB	0110	definNet.cpp:591	WARN	-
ODB	0111	definNonDefaultRule.cpp:64	WARN	-
ODB	0112	definNonDefaultRule.cpp:85	WARN	-
ODB	0113	definNonDefaultRule.cpp:101	WARN	-
ODB	0114	definNonDefaultRule.cpp:118	WARN	-
ODB	0115	definNonDefaultRule.cpp:134	WARN	-
ODB	0116	definNonDefaultRule.cpp:142	WARN	-
ODB	0117	definPin.cpp:118	WARN	-
ODB	0118	definPin.cpp:124	WARN	-
ODB	0119	definPin.cpp:202	WARN	-
ODB	0120	definPin.cpp:217	WARN	-
ODB	0121	definPin.cpp:237	WARN	-
ODB	0122	definPin.cpp:372	WARN	-
ODB	0123	definPin.cpp:385	WARN	-
ODB	0124	definReader.cpp:550	WARN	-
ODB	0125	definReader.cpp:1637	INFO	-
ODB	0126	definReader.cpp:1642	WARN	-
ODB	0127	definReader.cpp:1671	INFO	-
ODB	0128	definReader.cpp:361	INFO	-
ODB	0129	definReader.cpp:1675	WARN	-
ODB	0130	definReader.cpp:1680	INFO	-
ODB	0131	definReader.cpp:1686	INFO	-
ODB	0132	definReader.cpp:1697	INFO	-
ODB	0133	definReader.cpp:1704	INFO	-
ODB	0134	definReader.cpp:1717	INFO	-
ODB	0135	definReader.cpp:1730	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
ODB	0136	definNet.cpp:247	WARN	-
ODB	0137	definReader.cpp:1734	WARN	-
ODB	0138	definReader.cpp:1739	INFO	-
ODB	0139	definReader.cpp:1742	INFO	-
ODB	0140	definReader.cpp:1749	INFO	-
ODB	0141	definReader.cpp:1756	INFO	-
ODB	0142	definReader.cpp:1762	INFO	-
ODB	0143	definReader.cpp:1773	INFO	-
ODB	0144	definReader.cpp:1777	WARN	-
ODB	0145	definReader.cpp:1782	INFO	-
ODB	0146	definReader.cpp:1786	INFO	-
ODB	0147	definReader.cpp:1788	INFO	-
ODB	0148	definReader.cpp:1858	WARN	-
ODB	0149	definReader.cpp:1879	WARN	-
ODB	0150	definReader.cpp:1894	WARN	-
ODB	0151	definReader.cpp:1915	WARN	-
ODB	0152	definRegion.cpp:65	WARN	-
ODB	0153	dbPowerSwitch.cpp:270	ERROR	-
ODB	0154	dbPowerSwitch.cpp:280	ERROR	-
ODB	0155	definRow.cpp:111	WARN	-
ODB	0156	definSNet.cpp:123	WARN	-
ODB	0157	definSNet.cpp:178	WARN	-
ODB	0158	definSNet.cpp:188	WARN	-
ODB	0159	definSNet.cpp:243	WARN	-
ODB	0160	definSNet.cpp:262	WARN	-
ODB	0161	definSNet.cpp:297	WARN	-
ODB	0162	definSNet.cpp:315	WARN	-
ODB	0163	definSNet.cpp:442	WARN	-
ODB	0164	definSNet.cpp:499	WARN	-
ODB	0165	definTracks.cpp:70	WARN	-
ODB	0166	definVia.cpp:67	WARN	-
ODB	0167	definVia.cpp:80	WARN	-
ODB	0168	definVia.cpp:116	WARN	-
ODB	0169	definVia.cpp:125	WARN	-
ODB	0170	definVia.cpp:134	WARN	-
ODB	0171	definVia.cpp:231	WARN	-
ODB	0172	defout_impl.cpp:153	WARN	-
ODB	0173	defout_impl.cpp:930	WARN	-
ODB	0174	defout_impl.cpp:1662	WARN	-
ODB	0175	lefin.cpp:182	WARN	-
ODB	0176	lefin.cpp:204	WARN	-
ODB	0177	lefin.cpp:390	WARN	-
ODB	0178	lefin.cpp:410	WARN	-
ODB	0179	lefin.cpp:523	ERROR	-
ODB	0180	lefin.cpp:601	WARN	-
ODB	0181	lefin.cpp:622	WARN	-
ODB	0182	lefin.cpp:631	WARN	-
ODB	0183	lefin.cpp:829	ERROR	-
ODB	0184	lefin.cpp:1196	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
ODB	0185	lefin.cpp:1205	WARN	-
ODB	0186	lefin.cpp:1238	WARN	-
ODB	0187	lefin.cpp:1288	WARN	-
ODB	0188	lefin.cpp:1299	WARN	-
ODB	0189	lefin.cpp:1336	WARN	-
ODB	0190	lefin.cpp:1344	WARN	-
ODB	0191	lefin.cpp:1367	WARN	-
ODB	0192	lefin.cpp:1380	WARN	-
ODB	0193	lefin.cpp:1394	WARN	-
ODB	0194	lefin.cpp:1448	WARN	-
ODB	0195	lefin.cpp:1473	WARN	-
ODB	0196	lefin.cpp:1490	WARN	-
ODB	0197	lefin.cpp:1507	WARN	-
ODB	0198	lefin.cpp:1524	WARN	-
ODB	0199	lefin.cpp:1550	WARN	-
ODB	0200	lefin.cpp:1566	WARN	-
ODB	0201	lefin.cpp:1582	WARN	-
ODB	0202	lefin.cpp:1599	WARN	-
ODB	0203	lefin.cpp:1706	WARN	-
ODB	0204	lefin.cpp:1712	WARN	-
ODB	0205	lefin.cpp:1755	WARN	-
ODB	0206	lefin.cpp:1785	WARN	-
ODB	0207	lefin.cpp:1802	WARN	-
ODB	0209	lefin.cpp:1858	WARN	-
ODB	0210	lefin.cpp:1888	WARN	-
ODB	0211	lefin.cpp:1904	WARN	-
ODB	0212	lefin.cpp:1913	WARN	-
ODB	0213	lefin.cpp:1921	WARN	-
ODB	0214	lefin.cpp:1975	WARN	-
ODB	0215	lefin.cpp:1985	WARN	-
ODB	0216	lefin.cpp:2015	WARN	-
ODB	0217	lefin.cpp:2034	WARN	-
ODB	0218	lefin.cpp:2044	WARN	-
ODB	0219	dbPowerSwitch.cpp:297	ERROR	-
ODB	0220	dbPowerSwitch.cpp:307	ERROR	-
ODB	0221	lefin.cpp:2112	INFO	-
ODB	0222	lefin.cpp:2117	INFO	-
ODB	0223	lefin.cpp:2194	INFO	-
ODB	0224	lefin.cpp:2198	INFO	-
ODB	0225	lefin.cpp:2200	INFO	-
ODB	0226	lefin.cpp:2202	INFO	-
ODB	0228	lefin.cpp:2237	WARN	-
ODB	0229	lefin.cpp:2242	WARN	-
ODB	0230	lefin.cpp:2274	WARN	-
ODB	0231	lefin.cpp:2280	WARN	-
ODB	0240	reader.cpp:557	WARN	-
ODB	0241	create_box.cpp:154	WARN	-
ODB	0242	create_box.cpp:183	WARN	-
ODB	0243	create_box.cpp:186	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
ODB	0244	create_box.cpp:218	WARN	-
ODB	0245	create_box.cpp:247	WARN	-
ODB	0246	lefin.cpp:2185	ERROR	-
ODB	0247	definReader.cpp:943	WARN	-
ODB	0248	definReader.cpp:466	WARN	-
ODB	0249	definReader.cpp:1477	WARN	-
ODB	0250	definReader.cpp:1661	ERROR	-
ODB	0252	definReader.cpp:1683	INFO	-
ODB	0253	definReader.cpp:1693	INFO	-
ODB	0254	definReader.cpp:1711	INFO	-
ODB	0260	definReader.cpp:70	WARN	-
ODB	0261	definReader.cpp:334	WARN	-
ODB	0270	reader.cpp:548	WARN	-
ODB	0271	definReader.cpp:1867	WARN	-
ODB	0273	dbUtil.cpp:929	INFO	-
ODB	0274	create_box.cpp:59	WARN	-
ODB	0275	definReader.cpp:704	WARN	-
ODB	0276	create_box.cpp:252	INFO	-
ODB	0277	lefin.cpp:2129	WARN	-
ODB	0279	lefin.cpp:734	WARN	-
ODB	0280	lefin.cpp:2145	WARN	-
ODB	0282	dbTechLayer.cpp:1905	ERROR	-
ODB	0283	cdl.cpp:113	ERROR	-
ODB	0284	cdl.cpp:144	WARN	-
ODB	0285	cdl.cpp:148	WARN	-
ODB	0286	cdl.cpp:159	WARN	-
ODB	0287	cdl.cpp:220	ERROR	-
ODB	0288	lefin.cpp:2222	ERROR	-
ODB	0289	lefin.cpp:2302	ERROR	-
ODB	0292	lefin.cpp:2259	ERROR	-
ODB	0293	definReader.cpp:1320	WARN	-
ODB	0297	dbModule.cpp:186	ERROR	-
ODB	0298	dbModule.cpp:300	ERROR	-
ODB	0299	definVia.cpp:257	ERROR	-
ODB	0300	definVia.cpp:267	ERROR	-
ODB	0301	definVia.cpp:277	ERROR	-
ODB	0302	definVia.cpp:293	ERROR	-
ODB	0303	util.cpp:245	INFO	-
ODB	0304	definGroup.cpp:61	WARN	-
ODB	0305	definGroup.cpp:74	WARN	-
ODB	0306	definGroup.cpp:99	WARN	-
ODB	0307	dbBlock.cpp:3383	ERROR	-
ODB	0308	odb.tcl:10	ERROR	-
ODB	0309	odb.tcl:15	ERROR	-
ODB	0310	odb.tcl:27	ERROR	-
ODB	0311	odb.tcl:35	ERROR	-
ODB	0312	odb.tcl:38	ERROR	-
ODB	0313	odb.tcl:47	ERROR	-
ODB	0314	odb.tcl:59	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
ODB	0315	odb.tcl:196	ERROR	-
ODB	0316	odb.tcl:204	ERROR	-
ODB	0317	odb.tcl:211	ERROR	-
ODB	0318	odb.tcl:216	ERROR	-
ODB	0319	odb.tcl:223	ERROR	-
ODB	0320	odb.tcl:236	ERROR	-
ODB	0321	odb.tcl:241	ERROR	-
ODB	0322	odb.tcl:244	ERROR	-
ODB	0323	odb.tcl:257	ERROR	-
ODB	0324	odb.tcl:262	ERROR	-
ODB	0325	odb.tcl:265	ERROR	-
ODB	0326	odb.tcl:277	ERROR	-
ODB	0327	odb.tcl:282	ERROR	-
ODB	0328	odb.tcl:287	ERROR	-
ODB	0329	odb.tcl:293	ERROR	-
ODB	0330	odb.tcl:296	ERROR	-
ODB	0331	odb.tcl:299	ERROR	-
ODB	0332	odb.tcl:311	ERROR	-
ODB	0333	odb.tcl:316	ERROR	-
ODB	0334	odb.tcl:321	ERROR	-
ODB	0335	odb.tcl:327	ERROR	-
ODB	0336	odb.tcl:330	ERROR	-
ODB	0337	odb.tcl:333	ERROR	-
ODB	0338	odb.tcl:347	ERROR	-
ODB	0339	odb.tcl:352	ERROR	-
ODB	0340	odb.tcl:355	ERROR	-
ODB	0341	odb.tcl:360	ERROR	-
ODB	0342	odb.tcl:367	ERROR	-
ODB	0343	odb.tcl:374	ERROR	-
ODB	0344	odb.tcl:377	ERROR	-
ODB	0345	odb.tcl:392	ERROR	-
ODB	0346	odb.tcl:397	ERROR	-
ODB	0347	odb.tcl:400	ERROR	-
ODB	0348	odb.tcl:405	ERROR	-
ODB	0349	odb.tcl:409	ERROR	-
ODB	0350	odb.tcl:416	ERROR	-
ODB	0351	odb.tcl:423	ERROR	-
ODB	0352	odb.tcl:426	ERROR	-
ODB	0353	odb.tcl:438	ERROR	-
ODB	0354	odb.tcl:456	ERROR	-
ODB	0355	odb.tcl:519	ERROR	-
ODB	0356	lefin.cpp:2161	WARN	-
ODB	0357	cdl.cpp:214	WARN	-
ODB	0358	cdl.cpp:189	ERROR	-
ODB	0359	dbInst.cpp:478	ERROR	-
ODB	0360	dbInst.cpp:572	ERROR	-
ODB	0361	lefin.cpp:2175	WARN	-
ODB	0362	dbInst.cpp:1465	ERROR	-
ODB	0364	dbNet.cpp:2977	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
ODB	0367	dbModule.cpp:199	ERROR	-
ODB	0368	dbInst.cpp:1159	ERROR	-
ODB	0369	dbITerm.cpp:371	ERROR	-
ODB	0370	dbITerm.cpp:440	ERROR	-
ODB	0371	dbModule.cpp:234	ERROR	-
ODB	0372	dbITerm.cpp:451	ERROR	-
ODB	0373	dbITerm.cpp:379	ERROR	-
ODB	0374	dbBTerm.cpp:744	ERROR	-
ODB	0375	dbBTerm.cpp:473	ERROR	-
ODB	0376	dbBTerm.cpp:655	ERROR	-
ODB	0377	dbBTerm.cpp:438	ERROR	-
ODB	0378	dbBlock.cpp:3877	WARN	-
ODB	0379	dbGlobalConnect.cpp:307	WARN	-
ODB	0380	dbGlobalConnect.cpp:336	WARN	-
ODB	0381	dbBlock.cpp:3851	ERROR	-
ODB	0382	dbBlock.cpp:3855	WARN	-
ODB	0383	dbBlock.cpp:3923	WARN	-
ODB	0384	dbGlobalConnect.cpp:267	ERROR	-
ODB	0385	dbInst.cpp:1337	ERROR	-
ODB	0386	util.cpp:226	WARN	-
ODB	0387	definNonDefaultRule.cpp:224	WARN	-
ODB	0388	lefin.cpp:741	INFO	-
ODB	0389	dbUtil.cpp:92	ERROR	-
ODB	0390	tmg_conn.cpp:1213	ERROR	-
ODB	0391	tmg_conn.cpp:1243	ERROR	-
ODB	0392	tmg_conn.cpp:1279	ERROR	-
ODB	0393	tmg_conn.cpp:1750	ERROR	-
ODB	0394	lefin.cpp:1665	INFO	-
ODB	0395	tmg_conn.cpp:1565	WARN	-
ODB	0396	tmg_conn.cpp:1647	WARN	-
ODB	0398	dbUtil.cpp:735	WARN	-
ODB	0399	dbUtil.cpp:777	WARN	-
ODB	0400	dbUtil.cpp:1022	WARN	-
ODB	0401	dbUtil.cpp:1025	WARN	-
ODB	0402	dbUtil.cpp:1074	WARN	-
ODB	0403	dbUtil.cpp:1100	WARN	-
ODB	0404	dbUtil.cpp:1313	WARN	-
ODB	0405	dbUtil.cpp:1316	WARN	-
ODB	0406	dbUtil.cpp:1390	WARN	-
ODB	0407	dbUtil.cpp:1404	WARN	-
ODB	0408	dbUtil.cpp:1515	WARN	-
ODB	0409	dbUtil.cpp:1522	WARN	-
ODB	0410	dbUtil.cpp:1544	WARN	-
ODB	0411	dbUtil.cpp:1552	WARN	-
ODB	0412	dbUtil.cpp:1578	WARN	-
ODB	0413	dbUtil.cpp:1586	WARN	-
ODB	0417	dbUtil.cpp:131	WARN	-
ODB	0420	tmg_conn.cpp:606	ERROR	-
ODB	0421	definReader.cpp:1877	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
ODB	0422	definReader.cpp:1913	ERROR	-
ODB	0423	lefin.cpp:607	WARN	-
ODB	0424	parse.cpp:65	ERROR	-
ODB	0425	definComponentMaskShift.cpp:55	WARN	-
ODB	0428	parse.cpp:50	ERROR	-
ODB	0429	parse.cpp:412	ERROR	-
ODB	0430	dbBox.cpp:758	ERROR	-
ODB	0431	dbMaster.cpp:760	ERROR	-
ODB	0432	dbDatabase.cpp:446	ERROR	-
ODB	0433	dbITerm.cpp:386	ERROR	-
ODB	0436	dbInst.cpp:1423	ERROR	-
ODB	0437	definReader.cpp:983	WARN	-
ODB	1000	odb.tcl:69	WARN	-
ODB	1001	odb.tcl:137	WARN	-
ODB	1002	odb.tcl:140	WARN	-
ODB	1004	odb.tcl:152	ERROR	-
ODB	1005	odb.tcl:159	ERROR	-
ODB	1006	odb.tcl:164	ERROR	-
ODB	1007	odb.tcl:171	ERROR	-
ODB	1008	odb.tcl:179	ERROR	-
ODB	1009	odb.tcl:89	WARN	-
ODB	1100	dbAccessPoint.cpp:315	ERROR	-
ODB	1101	dbAccessPoint.cpp:335	ERROR	-
ODB	1102	dbWireCodec.cpp:630	ERROR	-
ODB	1103	dbWireCodec.cpp:654	ERROR	-
ODB	2000	lefin.cpp:1180	WARN	-
ORD	0001	OpenRoad.tcl:45	ERROR	-
ORD	0002	OpenRoad.tcl:48	ERROR	-
ORD	0003	OpenRoad.tcl:85	ERROR	-
ORD	0004	OpenRoad.tcl:88	ERROR	-
ORD	0005	OpenRoad.tcl:94	ERROR	-
ORD	0006	OpenRoad.tcl:124	ERROR	-
ORD	0007	OpenRoad.tcl:189	ERROR	-
ORD	0008	OpenRoad.tcl:192	ERROR	-
ORD	0013	OpenRoad.tcl:350	ERROR	-
ORD	0014	Metrics.tcl:64	ERROR	-
ORD	0015	OpenRoad.i:427	ERROR	-
ORD	0016	OpenRoad.tcl:104	ERROR	-
ORD	0017	Metrics.tcl:81	ERROR	-
ORD	0018	Metrics.tcl:98	ERROR	-
ORD	0019	Metrics.tcl:48	ERROR	-
ORD	0030	OpenRoad.cc:590	INFO	-
ORD	0031	OpenRoad.cc:576	WARN	-
ORD	0032	OpenRoad.cc:607	WARN	-
ORD	0033	OpenRoad.tcl:97	WARN	-
ORD	0034	OpenRoad.cc:412	INFO	-
ORD	0036	Design.cc:69	ERROR	-
ORD	0037	Design.cc:130	ERROR	-
ORD	0038	Main.cc:279	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
ORD	0039	Main.cc:283	WARN	-
ORD	0041	OpenRoad.tcl:296	ERROR	-
ORD	0042	OpenRoad.tcl:300	ERROR	-
ORD	0043	OpenRoad.tcl:308	ERROR	-
ORD	0044	OpenRoad.tcl:315	WARN	-
ORD	0045	OpenRoad.tcl:337	ERROR	-
ORD	0046	OpenRoad.tcl:329	WARN	-
ORD	0047	OpenRoad.cc:466	ERROR	-
ORD	0048	OpenRoad.cc:315	INFO	-
ORD	0049	OpenRoad.i:477	ERROR	-
ORD	0050	OpenRoad.i:488	ERROR	-
ORD	0051	OpenRoad.cc:293	ERROR	-
ORD	0052	OpenRoad.i:336	ERROR	-
ORD	0053	OpenRoad.cc:395	ERROR	-
ORD	0054	OpenRoad.cc:478	ERROR	-
ORD	0101	Design.cc:85	ERROR	-
ORD	0102	Design.cc:91	ERROR	-
ORD	0105	OpenRoad.cc:512	ERROR	-
ORD	0201	Resizer.tcl:48	ERROR	-
ORD	0202	Resizer.tcl:57	ERROR	-
ORD	0203	Resizer.tcl:61	ERROR	-
ORD	0204	Resizer.tcl:65	ERROR	-
ORD	0205	Resizer.tcl:98	ERROR	-
ORD	0206	Resizer.tcl:102	WARN	-
ORD	0208	Resizer.tcl:120	ERROR	-
ORD	0209	Resizer.tcl:123	ERROR	-
ORD	1009	OpenRoad.tcl:210	ERROR	-
ORD	1010	OpenRoad.tcl:213	ERROR	-
ORD	1011	OpenRoad.tcl:219	ERROR	-
ORD	1012	OpenRoad.tcl:225	ERROR	-
ORD	1013	OpenRoad.tcl:172	ERROR	-
ORD	1050	OpenRoad.tcl:155	ERROR	-
ORD	2001	dbNetwork.cc:918	WARN	-
ORD	2002	dbNetwork.cc:1041	WARN	-
ORD	2003	dbNetwork.cc:1218	CRITICAL	-
ORD	2004	dbNetwork.cc:1273	CRITICAL	-
ORD	2005	dbNetwork.cc:1278	CRITICAL	-
ORD	2006	dbNetwork.cc:1342	CRITICAL	-
ORD	2007	dbNetwork.cc:1399	CRITICAL	-
ORD	2008	dbNetwork.cc:1471	CRITICAL	-
ORD	2009	dbReadVerilog.tcl:51	ERROR	-
ORD	2010	dbReadVerilog.tcl:58	ERROR	-
ORD	2011	dbReadVerilog.cc:426	WARN	-
ORD	2012	dbReadVerilog.cc:432	WARN	-
ORD	2013	dbReadVerilog.cc:294	WARN	-
ORD	2014	dbReadVerilog.cc:276	WARN	-
ORD	2015	dbReadVerilog.cc:303	WARN	-
ORD	2016	dbNetwork.cc:1314	CRITICAL	-
PAD	0001	ICeWall.cpp:662	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAD	0002	ICeWall.cpp:1133	ERROR	-
PAD	0003	RDLRouter.cpp:160	WARN	-
PAD	0004	RDLRouter.cpp:174	WARN	-
PAD	0005	RDLRouter.cpp:232	INFO	-
PAD	0006	RDLRouter.cpp:278	WARN	-
PAD	0007	RDLRouter.cpp:302	ERROR	-
PAD	0008	RDLRouter.cpp:349	ERROR	-
PAD	0009	RDLRouter.cpp:474	ERROR	-
PAD	0010	RDLRouter.cpp:1206	ERROR	-
PAD	0011	ICeWall.cpp:76	ERROR	-
PAD	0012	ICeWall.cpp:90	ERROR	-
PAD	0013	ICeWall.cpp:458	ERROR	-
PAD	0014	ICeWall.cpp:307	ERROR	-
PAD	0015	ICeWall.cpp:310	ERROR	-
PAD	0016	ICeWall.cpp:313	ERROR	-
PAD	0018	ICeWall.cpp:492	ERROR	-
PAD	0019	ICeWall.cpp:508	ERROR	-
PAD	0020	ICeWall.cpp:696	ERROR	-
PAD	0021	ICeWall.cpp:874	ERROR	-
PAD	0022	ICeWall.cpp:1328	ERROR	-
PAD	0023	ICeWall.cpp:73	ERROR	-
PAD	0024	ICeWall.cpp:196	ERROR	-
PAD	0025	ICeWall.cpp:201	ERROR	-
PAD	0026	ICeWall.cpp:778	ERROR	-
PAD	0027	ICeWall.cpp:899	ERROR	-
PAD	0028	ICeWall.cpp:451	ERROR	-
PAD	0029	ICeWall.cpp:569	ERROR	-
PAD	0030	ICeWall.cpp:854	ERROR	-
PAD	0031	ICeWall.cpp:935	WARN	-
PAD	0032	ICeWall.cpp:929	ERROR	-
PAD	0033	ICeWall.cpp:159	WARN	-
PAD	0034	ICeWall.cpp:166	ERROR	-
PAD	0035	ICeWall.cpp:222	ERROR	-
PAD	0036	ICeWall.cpp:231	ERROR	-
PAD	0037	RDLRouter.cpp:1306	ERROR	-
PAD	0100	pad.tcl:440	ERROR	-
PAD	0101	pad.tcl:448	ERROR	-
PAD	0102	pad.tcl:456	ERROR	-
PAD	0103	pad.tcl:464	ERROR	-
PAD	0104	pad.tcl:472	ERROR	-
PAD	0105	pad.tcl:386	ERROR	-
PAD	0106	pad.i:160	ERROR	-
PAD	0107	pad.tcl:392	ERROR	-
PAD	0108	pad.tcl:399	ERROR	-
PAD	0109	pad.tcl:480	ERROR	-
PAD	0110	pad.tcl:485	ERROR	-
PAD	0111	pad.tcl:490	ERROR	-
PAD	0112	pad.tcl:156	WARN	-
PAD	0113	pad.tcl:114	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAD	0114	pad.tcl:120	ERROR	-
PAD	0115	ICeWall.cpp:1039	ERROR	-
PAD	0116	ICeWall.cpp:177	INFO	-
PAD	0117	pad.tcl:303	ERROR	-
PAD	0118	ICeWall.cpp:499	ERROR	-
PAD	0119	ICeWall.cpp:633	ERROR	-
PAD	0232	grid.cpp:1540	WARN	-
PAD	9001	ICeWall.tcl:1925	ERROR	-
PAD	9002	ICeWall.tcl:1721	ERROR	-
PAD	9004	ICeWall.tcl:2508	WARN	-
PAD	9005	ICeWall.tcl:412	ERROR	-
PAD	9006	ICeWall.tcl:458	ERROR	-
PAD	9007	ICeWall.tcl:1895	ERROR	-
PAD	9008	ICeWall.tcl:1134	ERROR	-
PAD	9009	ICeWall.tcl:5020	ERROR	-
PAD	9010	ICeWall.tcl:5049	ERROR	-
PAD	9011	ICeWall.tcl:1683	WARN	-
PAD	9012	ICeWall.tcl:2552	WARN	-
PAD	9014	ICeWall.tcl:4671	ERROR	-
PAD	9015	ICeWall.tcl:4344	ERROR	-
PAD	9016	ICeWall.tcl:768	ERROR	-
PAD	9017	ICeWall.tcl:2038	ERROR	-
PAD	9018	ICeWall.tcl:4696	WARN	-
PAD	9019	ICeWall.tcl:2599	WARN	-
PAD	9021	ICeWall.tcl:1341	ERROR	-
PAD	9022	ICeWall.tcl:1323	ERROR	-
PAD	9023	ICeWall.tcl:1328	ERROR	-
PAD	9024	ICeWall.tcl:495	ERROR	-
PAD	9025	ICeWall.tcl:566	ERROR	-
PAD	9026	ICeWall.tcl:506	ERROR	-
PAD	9027	ICeWall.tcl:3276	ERROR	-
PAD	9028	ICeWall.tcl:3296	ERROR	-
PAD	9029	ICeWall.tcl:4748	ERROR	-
PAD	9030	ICeWall.tcl:4816	ERROR	-
PAD	9031	ICeWall.tcl:671	ERROR	-
PAD	9032	ICeWall.tcl:967	ERROR	-
PAD	9033	ICeWall.tcl:1240	ERROR	-
PAD	9034	ICeWall.tcl:2727	ERROR	-
PAD	9035	ICeWall.tcl:2813	ERROR	-
PAD	9036	ICeWall.tcl:2893	ERROR	-
PAD	9037	ICeWall.tcl:2865	ERROR	-
PAD	9038	ICeWall.tcl:2904	ERROR	-
PAD	9039	ICeWall.tcl:2916	ERROR	-
PAD	9040	ICeWall.tcl:2929	ERROR	-
PAD	9041	ICeWall.tcl:685	ERROR	-
PAD	9042	ICeWall.tcl:2095	WARN	-
PAD	9043	ICeWall.tcl:2098	ERROR	-
PAD	9044	ICeWall.tcl:2230	ERROR	-
PAD	9045	ICeWall.tcl:2246	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAD	9046	ICeWall.tcl:2285	WARN	-
PAD	9047	ICeWall.tcl:2290	WARN	-
PAD	9048	ICeWall.tcl:2654	WARN	-
PAD	9049	ICeWall.tcl:1143	ERROR	-
PAD	9050	ICeWall.tcl:4661	WARN	-
PAD	9051	ICeWall.tcl:2539	INFO	-
PAD	9052	ICeWall.tcl:3831	INFO	-
PAD	9053	ICeWall.tcl:4702	INFO	-
PAD	9054	ICeWall.tcl:374	ERROR	-
PAD	9055	ICeWall.tcl:377	ERROR	-
PAD	9056	ICeWall.tcl:420	ERROR	-
PAD	9057	ICeWall.tcl:423	ERROR	-
PAD	9058	ICeWall.tcl:469	ERROR	-
PAD	9059	ICeWall.tcl:473	ERROR	-
PAD	9060	ICeWall.tcl:585	ERROR	-
PAD	9061	ICeWall.tcl:716	ERROR	-
PAD	9062	ICeWall.tcl:727	ERROR	-
PAD	9063	ICeWall.tcl:864	ERROR	-
PAD	9064	ICeWall.tcl:870	ERROR	-
PAD	9065	ICeWall.tcl:912	ERROR	-
PAD	9066	ICeWall.tcl:1012	ERROR	-
PAD	9070	ICeWall.tcl:1189	ERROR	-
PAD	9071	ICeWall.tcl:1203	ERROR	-
PAD	9072	ICeWall.tcl:1398	ERROR	-
PAD	9073	ICeWall.tcl:1402	ERROR	-
PAD	9074	ICeWall.tcl:1405	ERROR	-
PAD	9075	ICeWall.tcl:1442	ERROR	-
PAD	9076	ICeWall.tcl:1445	ERROR	-
PAD	9077	ICeWall.tcl:1448	ERROR	-
PAD	9078	ICeWall.tcl:2566	ERROR	-
PAD	9079	ICeWall.tcl:2968	ERROR	-
PAD	9080	ICeWall.tcl:4428	ERROR	-
PAD	9081	ICeWall.tcl:4438	ERROR	-
PAD	9082	ICeWall.tcl:4563	ERROR	-
PAD	9083	ICeWall.tcl:4567	ERROR	-
PAD	9084	ICeWall.tcl:4721	ERROR	-
PAD	9085	ICeWall.tcl:4725	ERROR	-
PAD	9086	ICeWall.tcl:4881	ERROR	-
PAD	9087	ICeWall.tcl:4885	ERROR	-
PAD	9091	ICeWall.tcl:5293	ERROR	-
PAD	9092	ICeWall.tcl:5291	ERROR	-
PAD	9093	ICeWall.tcl:5307	ERROR	-
PAD	9094	ICeWall.tcl:5323	ERROR	-
PAD	9095	ICeWall.tcl:5333	ERROR	-
PAD	9096	ICeWall.tcl:1145	ERROR	-
PAD	9097	ICeWall.tcl:1147	ERROR	-
PAD	9098	ICeWall.tcl:5336	ERROR	-
PAD	9099	ICeWall.tcl:5434	ERROR	-
PAD	9100	ICeWall.tcl:5441	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAD	9101	ICeWall.tcl:5444	ERROR	-
PAD	9102	ICeWall.tcl:5449	ERROR	-
PAD	9103	ICeWall.tcl:5454	ERROR	-
PAD	9104	ICeWall.tcl:5457	ERROR	-
PAD	9105	ICeWall.tcl:5471	ERROR	-
PAD	9106	ICeWall.tcl:5480	ERROR	-
PAD	9109	ICeWall.tcl:5858	ERROR	-
PAD	9110	ICeWall.tcl:5876	ERROR	-
PAD	9111	ICeWall.tcl:5779	ERROR	-
PAD	9112	ICeWall.tcl:5909	ERROR	-
PAD	9113	ICeWall.tcl:6538	ERROR	-
PAD	9114	ICeWall.tcl:604	ERROR	-
PAD	9115	ICeWall.tcl:624	ERROR	-
PAD	9116	ICeWall.tcl:787	ERROR	-
PAD	9117	ICeWall.tcl:803	ERROR	-
PAD	9119	ICeWall.tcl:808	ERROR	-
PAD	9120	ICeWall.tcl:1167	ERROR	-
PAD	9121	ICeWall.tcl:1170	ERROR	-
PAD	9122	ICeWall.tcl:5539	ERROR	-
PAD	9123	ICeWall.tcl:5925	ERROR	-
PAD	9124	ICeWall.tcl:5955	ERROR	-
PAD	9125	ICeWall.tcl:5962	ERROR	-
PAD	9126	ICeWall.tcl:5968	ERROR	-
PAD	9127	ICeWall.tcl:5987	ERROR	-
PAD	9128	ICeWall.tcl:5994	ERROR	-
PAD	9129	ICeWall.tcl:6012	ERROR	-
PAD	9130	ICeWall.tcl:6018	ERROR	-
PAD	9131	ICeWall.tcl:6031	ERROR	-
PAD	9132	ICeWall.tcl:6037	ERROR	-
PAD	9133	ICeWall.tcl:6051	ERROR	-
PAD	9134	ICeWall.tcl:6062	ERROR	-
PAD	9135	ICeWall.tcl:6066	WARN	-
PAD	9136	ICeWall.tcl:6077	ERROR	-
PAD	9137	ICeWall.tcl:6081	ERROR	-
PAD	9140	ICeWall.tcl:5949	ERROR	-
PAD	9141	ICeWall.tcl:2025	ERROR	-
PAD	9142	ICeWall.tcl:5601	ERROR	-
PAD	9143	ICeWall.tcl:5606	ERROR	-
PAD	9144	ICeWall.tcl:5626	ERROR	-
PAD	9145	ICeWall.tcl:5631	ERROR	-
PAD	9146	ICeWall.tcl:5730	ERROR	-
PAD	9147	ICeWall.tcl:5707	ERROR	-
PAD	9159	ICeWall.tcl:6005	INFO	-
PAD	9160	ICeWall.tcl:5716	ERROR	-
PAD	9161	ICeWall.tcl:1060	ERROR	-
PAD	9162	ICeWall.tcl:2977	ERROR	-
PAD	9163	ICeWall.tcl:6110	ERROR	-
PAD	9164	ICeWall.tcl:6113	ERROR	-
PAD	9165	ICeWall.tcl:6165	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAD	9166	ICeWall.tcl:6178	ERROR	-
PAD	9167	ICeWall.tcl:6184	ERROR	-
PAD	9168	ICeWall.tcl:6191	ERROR	-
PAD	9169	ICeWall.tcl:6197	ERROR	-
PAD	9170	ICeWall.tcl:6207	ERROR	-
PAD	9171	ICeWall.tcl:6213	ERROR	-
PAD	9173	ICeWall.tcl:6228	ERROR	-
PAD	9174	ICeWall.tcl:6312	ERROR	-
PAD	9175	ICeWall.tcl:6323	ERROR	-
PAD	9176	ICeWall.tcl:6343	ERROR	-
PAD	9177	ICeWall.tcl:6347	ERROR	-
PAD	9178	ICeWall.tcl:6363	ERROR	-
PAD	9179	ICeWall.tcl:6371	ERROR	-
PAD	9180	ICeWall.tcl:6410	ERROR	-
PAD	9181	ICeWall.tcl:6426	ERROR	-
PAD	9182	ICeWall.tcl:6437	ERROR	-
PAD	9183	ICeWall.tcl:6478	ERROR	-
PAD	9184	ICeWall.tcl:6492	ERROR	-
PAD	9185	ICeWall.tcl:6512	ERROR	-
PAD	9187	ICeWall.tcl:365	ERROR	-
PAD	9188	ICeWall.tcl:6616	ERROR	-
PAD	9189	ICeWall.tcl:6625	ERROR	-
PAD	9190	ICeWall.tcl:6632	ERROR	-
PAD	9191	ICeWall.tcl:6640	ERROR	-
PAD	9192	ICeWall.tcl:6643	ERROR	-
PAD	9193	ICeWall.tcl:6650	ERROR	-
PAD	9194	ICeWall.tcl:6653	ERROR	-
PAD	9195	ICeWall.tcl:6656	ERROR	-
PAD	9196	ICeWall.tcl:6659	ERROR	-
PAD	9197	ICeWall.tcl:6666	ERROR	-
PAD	9198	ICeWall.tcl:6671	ERROR	-
PAD	9199	ICeWall.tcl:6677	ERROR	-
PAD	9200	ICeWall.tcl:5901	ERROR	-
PAD	9201	ICeWall.tcl:6389	ERROR	-
PAD	9202	ICeWall.tcl:6397	ERROR	-
PAD	9203	ICeWall.tcl:358	ERROR	-
PAD	9204	ICeWall.tcl:361	ERROR	-
PAD	9205	ICeWall.tcl:5685	ERROR	-
PAD	9207	ICeWall.tcl:4779	ERROR	-
PAD	9208	ICeWall.tcl:205	ERROR	-
PAD	9209	ICeWall.tcl:5789	ERROR	-
PAD	9210	ICeWall.tcl:5794	ERROR	-
PAD	9211	ICeWall.tcl:5807	ERROR	-
PAD	9212	ICeWall.tcl:5818	ERROR	-
PAD	9213	ICeWall.tcl:5821	ERROR	-
PAD	9214	ICeWall.tcl:5833	ERROR	-
PAD	9215	ICeWall.tcl:5840	ERROR	-
PAD	9216	ICeWall.tcl:5880	ERROR	-
PAD	9217	ICeWall.tcl:2381	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAD	9218	ICeWall.tcl:51	ERROR	-
PAD	9219	ICeWall.tcl:124	ERROR	-
PAD	9220	ICeWall.tcl:195	ERROR	-
PAD	9221	ICeWall.tcl:235	ERROR	-
PAD	9222	ICeWall.tcl:250	ERROR	-
PAD	9223	ICeWall.tcl:2353	ERROR	-
PAD	9224	ICeWall.tcl:231	ERROR	-
PAD	9225	ICeWall.tcl:191	ERROR	-
PAD	9226	ICeWall.tcl:116	ERROR	-
PAD	9227	ICeWall.tcl:243	ERROR	-
PAD	9228	ICeWall.tcl:6604	ERROR	-
PAD	9229	ICeWall.tcl:5421	ERROR	-
PAD	9230	ICeWall.tcl:5424	ERROR	-
PAD	9231	ICeWall.tcl:60	ERROR	-
PAD	9232	ICeWall.tcl:83	ERROR	-
PAD	9233	ICeWall.tcl:72	ERROR	-
PAD	9234	ICeWall.tcl:75	ERROR	-
PAD	9235	ICeWall.tcl:3736	ERROR	-
PAD	9236	ICeWall.tcl:3863	ERROR	-
PAD	9237	ICeWall.tcl:68	ERROR	-
PAD	9238	ICeWall.tcl:3709	ERROR	-
PAD	9239	ICeWall.tcl:5351	ERROR	-
PAD	9240	ICeWall.tcl:5344	ERROR	-
PAD	9241	ICeWall.tcl:5375	ERROR	-
PAD	9242	ICeWall.tcl:5379	ERROR	-
PAD	9243	ICeWall.tcl:5408	ERROR	-
PAD	9244	ICeWall.tcl:5411	ERROR	-
PAD	9245	ICeWall.tcl:5414	ERROR	-
PAD	9246	ICeWall.tcl:5777	INFO	-
PAD	9247	ICeWall.tcl:4221	ERROR	-
PAD	9248	ICeWall.tcl:4272	ERROR	-
PAD	9249	ICeWall.tcl:4240	ERROR	-
PAD	9250	ICeWall.tcl:4033	ERROR	-
PAD	9251	ICeWall.tcl:1123	WARN	-
PAD	9252	ICeWall.tcl:2642	ERROR	-
PAD	9253	ICeWall.tcl:3814	ERROR	-
PAD	9254	ICeWall.tcl:5081	WARN	-
PAD	9255	ICeWall.tcl:5083	WARN	-
PAD	9256	ICeWall.tcl:5086	ERROR	-
PAD	9257	ICeWall.tcl:2621	ERROR	-
PAD	9258	ICeWall.tcl:5749	ERROR	-
PAD	9259	ICeWall.tcl:5739	ERROR	-
PAD	9260	ICeWall.tcl:3465	ERROR	-
PAD	9261	ICeWall.tcl:3500	ERROR	-
PAD	9262	ICeWall.tcl:3441	WARN	-
PAD	9263	ICeWall.tcl:3445	WARN	-
PAD	9264	ICeWall.tcl:3587	WARN	-
PAD	9265	ICeWall.tcl:3625	WARN	-
PAD	9266	ICeWall.tcl:3627	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAD	9267	ICeWall.tcl:3656	WARN	-
PAD	9268	ICeWall.tcl:3658	WARN	-
PAD	9269	ICeWall.tcl:3665	WARN	-
PAD	9270	ICeWall.tcl:3667	WARN	-
PAD	9271	ICeWall.tcl:3677	WARN	-
PAR	0001	Coarsener.cpp:155	INFO	-
PAR	0002	TritonPart.cpp:256	INFO	-
PAR	0003	TritonPart.cpp:257	INFO	-
PAR	0004	TritonPart.cpp:258	INFO	-
PAR	0005	TritonPart.cpp:259	INFO	-
PAR	0006	TritonPart.cpp:260	INFO	-
PAR	0007	TritonPart.cpp:261	INFO	-
PAR	0008	TritonPart.cpp:262	INFO	-
PAR	0009	TritonPart.cpp:263	INFO	-
PAR	0010	TritonPart.cpp:264	INFO	-
PAR	0011	TritonPart.cpp:266	INFO	-
PAR	0012	TritonPart.cpp:269	INFO	-
PAR	0013	TritonPart.cpp:272	INFO	-
PAR	0014	TritonPart.cpp:275	INFO	-
PAR	0015	PartitionMgr.cpp:771	WARN	-
PAR	0016	TritonPart.cpp:378	INFO	-
PAR	0017	TritonPart.cpp:379	INFO	-
PAR	0018	TritonPart.cpp:380	INFO	-
PAR	0019	TritonPart.cpp:381	INFO	-
PAR	0020	TritonPart.cpp:382	INFO	-
PAR	0021	TritonPart.cpp:383	INFO	-
PAR	0022	PartitionMgr.cpp:860	ERROR	-
PAR	0023	TritonPart.cpp:385	INFO	-
PAR	0024	TritonPart.cpp:386	INFO	-
PAR	0025	TritonPart.cpp:387	INFO	-
PAR	0026	TritonPart.cpp:389	INFO	-
PAR	0027	TritonPart.cpp:398	INFO	-
PAR	0028	TritonPart.cpp:401	INFO	-
PAR	0029	TritonPart.cpp:404	INFO	-
PAR	0030	TritonPart.cpp:407	INFO	-
PAR	0031	TritonPart.cpp:535	INFO	-
PAR	0032	TritonPart.cpp:536	INFO	-
PAR	0033	TritonPart.cpp:537	INFO	-
PAR	0034	TritonPart.cpp:538	INFO	-
PAR	0035	TritonPart.cpp:539	INFO	-
PAR	0036	TritonPart.cpp:540	INFO	-
PAR	0037	TritonPart.cpp:541	INFO	-
PAR	0038	TritonPart.cpp:542	INFO	-
PAR	0039	TritonPart.cpp:544	INFO	-
PAR	0040	TritonPart.cpp:547	INFO	-
PAR	0041	TritonPart.cpp:550	INFO	-
PAR	0042	TritonPart.cpp:553	INFO	-
PAR	0043	TritonPart.cpp:611	INFO	-
PAR	0044	TritonPart.cpp:623	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAR	0045	TritonPart.cpp:640	INFO	-
PAR	0046	TritonPart.cpp:653	INFO	-
PAR	0047	TritonPart.cpp:654	INFO	-
PAR	0048	TritonPart.cpp:655	INFO	-
PAR	0049	TritonPart.cpp:656	INFO	-
PAR	0050	TritonPart.cpp:657	INFO	-
PAR	0051	partitionmgr.tcl:1016	ERROR	-
PAR	0052	TritonPart.cpp:755	INFO	-
PAR	0053	TritonPart.cpp:756	INFO	-
PAR	0054	TritonPart.cpp:757	INFO	-
PAR	0055	TritonPart.cpp:758	INFO	-
PAR	0056	TritonPart.cpp:759	INFO	-
PAR	0057	TritonPart.cpp:760	INFO	-
PAR	0058	TritonPart.cpp:761	INFO	-
PAR	0059	TritonPart.cpp:762	INFO	-
PAR	0060	TritonPart.cpp:763	INFO	-
PAR	0061	TritonPart.cpp:764	INFO	-
PAR	0062	TritonPart.cpp:766	INFO	-
PAR	0063	TritonPart.cpp:775	INFO	-
PAR	0064	TritonPart.cpp:778	INFO	-
PAR	0065	TritonPart.cpp:781	INFO	-
PAR	0066	TritonPart.cpp:784	INFO	-
PAR	0067	TritonPart.cpp:787	INFO	-
PAR	0068	TritonPart.cpp:791	INFO	-
PAR	0069	TritonPart.cpp:836	INFO	-
PAR	0070	TritonPart.cpp:848	INFO	-
PAR	0071	PartitionMgr.cpp:870	ERROR	-
PAR	0072	PartitionMgr.cpp:838	ERROR	-
PAR	0073	PartitionMgr.cpp:847	ERROR	-
PAR	0074	PartitionMgr.cpp:881	ERROR	-
PAR	0075	TritonPart.cpp:874	INFO	-
PAR	0076	TritonPart.cpp:875	INFO	-
PAR	0077	TritonPart.cpp:1780	INFO	-
PAR	0078	TritonPart.cpp:1792	INFO	-
PAR	0079	TritonPart.cpp:1809	INFO	-
PAR	0080	TritonPart.cpp:1814	INFO	-
PAR	0081	TritonPart.cpp:1815	INFO	-
PAR	0082	TritonPart.cpp:1816	INFO	-
PAR	0083	TritonPart.cpp:1817	INFO	-
PAR	0084	TritonPart.cpp:1819	INFO	-
PAR	0085	TritonPart.cpp:1820	INFO	-
PAR	0086	TritonPart.cpp:1824	INFO	-
PAR	0087	TritonPart.cpp:1825	INFO	-
PAR	0088	TritonPart.cpp:1827	INFO	-
PAR	0089	TritonPart.cpp:1828	INFO	-
PAR	0090	TritonPart.cpp:1829	INFO	-
PAR	0091	TritonPart.cpp:1830	INFO	-
PAR	0092	TritonPart.cpp:1833	INFO	-
PAR	0093	TritonPart.cpp:1834	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAR	0094	TritonPart.cpp:1837	INFO	-
PAR	0095	TritonPart.cpp:1838	INFO	-
PAR	0096	TritonPart.cpp:1840	INFO	-
PAR	0097	TritonPart.cpp:1841	INFO	-
PAR	0098	TritonPart.cpp:1842	INFO	-
PAR	0099	TritonPart.cpp:1843	INFO	-
PAR	0100	TritonPart.cpp:1844	INFO	-
PAR	0101	TritonPart.cpp:1845	INFO	-
PAR	0102	TritonPart.cpp:377	INFO	-
PAR	0103	TritonPart.cpp:384	INFO	-
PAR	0104	TritonPart.cpp:754	INFO	-
PAR	0105	TritonPart.cpp:865	INFO	-
PAR	0106	TritonPart.cpp:871	INFO	-
PAR	0107	TritonPart.cpp:872	INFO	-
PAR	0108	TritonPart.cpp:873	INFO	-
PAR	0109	TritonPart.cpp:2024	INFO	-
PAR	0110	TritonPart.cpp:472	INFO	-
PAR	0111	Evaluator.cpp:113	WARN	-
PAR	0112	Evaluator.cpp:130	WARN	-
PAR	0113	Evaluator.cpp:178	WARN	-
PAR	0114	Evaluator.cpp:195	WARN	-
PAR	0115	ILPRefine.cpp:134	WARN	-
PAR	0116	Partitioner.cpp:276	WARN	-
PAR	0117	Partitioner.cpp:278	WARN	-
PAR	0118	Refiner.cpp:119	INFO	-
PAR	0119	TritonPart.cpp:282	WARN	-
PAR	0120	TritonPart.cpp:571	WARN	-
PAR	0121	TritonPart.cpp:603	WARN	-
PAR	0124	TritonPart.cpp:617	WARN	-
PAR	0125	TritonPart.cpp:630	WARN	-
PAR	0126	TritonPart.cpp:828	WARN	-
PAR	0127	TritonPart.cpp:842	WARN	-
PAR	0128	TritonPart.cpp:855	WARN	-
PAR	0129	TritonPart.cpp:1098	WARN	-
PAR	0130	TritonPart.cpp:1116	WARN	-
PAR	0132	TritonPart.cpp:1203	WARN	-
PAR	0133	TritonPart.cpp:1356	WARN	-
PAR	0134	TritonPart.cpp:1383	WARN	-
PAR	0135	TritonPart.cpp:1410	WARN	-
PAR	0136	TritonPart.cpp:1548	WARN	-
PAR	0137	TritonPart.cpp:1733	WARN	-
PAR	0138	TritonPart.cpp:1738	WARN	-
PAR	0139	TritonPart.cpp:1772	WARN	-
PAR	0140	TritonPart.cpp:1799	WARN	-
PAR	0141	TritonPart.cpp:1786	WARN	-
PAR	0142	Evaluator.cpp:290	WARN	-
PAR	0143	Evaluator.cpp:249	INFO	-
PAR	0144	Evaluator.cpp:251	INFO	-
PAR	0145	Evaluator.cpp:255	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAR	0146	Evaluator.cpp:259	INFO	-
PAR	0147	Evaluator.cpp:263	INFO	-
PAR	0148	Evaluator.cpp:267	INFO	-
PAR	0149	Evaluator.cpp:272	INFO	-
PAR	0150	Evaluator.cpp:276	INFO	-
PAR	0151	Multilevel.cpp:120	INFO	-
PAR	0152	Multilevel.cpp:129	INFO	-
PAR	0153	Multilevel.cpp:141	INFO	-
PAR	0154	Multilevel.cpp:218	INFO	-
PAR	0155	Multilevel.cpp:444	INFO	-
PAR	0156	Multilevel.cpp:449	INFO	-
PAR	0157	Multilevel.cpp:625	INFO	-
PAR	0158	Multilevel.cpp:652	INFO	-
PAR	0159	Partitioner.cpp:62	INFO	-
PAR	0160	Partitioner.cpp:69	INFO	-
PAR	0161	Partitioner.cpp:268	INFO	-
PAR	0162	Partitioner.cpp:270	INFO	-
PAR	0163	Refiner.cpp:93	INFO	-
PAR	0164	Refiner.cpp:99	INFO	-
PAR	0165	Refiner.cpp:107	INFO	-
PAR	0166	Refiner.cpp:108	INFO	-
PAR	0167	TritonPart.cpp:237	INFO	-
PAR	0168	TritonPart.cpp:339	INFO	-
PAR	0169	TritonPart.cpp:514	INFO	-
PAR	0170	TritonPart.cpp:714	INFO	-
PAR	0171	TritonPart.cpp:1221	INFO	-
PAR	0172	TritonPart.cpp:1222	INFO	-
PAR	0173	TritonPart.cpp:1224	INFO	-
PAR	0174	TritonPart.cpp:1532	INFO	-
PAR	0175	TritonPart.cpp:1533	INFO	-
PAR	0176	TritonPart.cpp:1535	INFO	-
PAR	0177	TritonPart.cpp:1537	INFO	-
PAR	0178	TritonPart.cpp:1680	INFO	-
PAR	0179	TritonPart.cpp:1683	INFO	-
PAR	0180	TritonPart.cpp:1694	INFO	-
PAR	0181	TritonPart.cpp:1704	INFO	-
PAR	0350	TritonPart.cpp:583	WARN	-
PAR	0351	TritonPart.cpp:808	WARN	-
PAR	0352	TritonPart.cpp:1752	WARN	-
PAR	0353	TritonPart.cpp:1759	WARN	-
PAR	0354	TritonPart.cpp:590	WARN	-
PAR	0355	TritonPart.cpp:815	WARN	-
PAR	0924	partitionmgr.tcl:114	ERROR	-
PAR	0925	partitionmgr.tcl:352	ERROR	-
PAR	2500	TritonPart.cpp:1008	ERROR	-
PAR	2501	TritonPart.cpp:1091	ERROR	-
PAR	2502	TritonPart.cpp:1108	ERROR	-
PAR	2503	TritonPart.cpp:1125	ERROR	-
PAR	2504	TritonPart.cpp:1148	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PAR	2511	TritonPart.cpp:559	ERROR	-
PAR	2514	TritonPart.cpp:910	ERROR	-
PAR	2677	TritonPart.cpp:1506	ERROR	-
PDN	0001	grid.cpp:134	INFO	-
PDN	0100	domain.cpp:260	ERROR	-
PDN	0101	domain.cpp:282	INFO	-
PDN	0102	domain.cpp:291	INFO	-
PDN	0103	domain.cpp:84	ERROR	-
PDN	0104	domain.cpp:87	ERROR	-
PDN	0105	rings.cpp:123	WARN	-
PDN	0106	grid_component.cpp:364	ERROR	-
PDN	0107	grid_component.cpp:376	ERROR	-
PDN	0108	grid_component.cpp:446	ERROR	-
PDN	0109	straps.cpp:473	ERROR	-
PDN	0110	via.cpp:938	WARN	-
PDN	0113	PdnGen.tcl:572	ERROR	-
PDN	0114	grid_component.cpp:424	ERROR	-
PDN	0115	sroute.cpp:95	ERROR	-
PDN	0116	sroute.cpp:151	ERROR	-
PDN	0174	PdnGen.tcl:6608	ERROR	-
PDN	0175	straps.cpp:102	ERROR	-
PDN	0178	straps.cpp:2004	WARN	-
PDN	0179	straps.cpp:2013	ERROR	-
PDN	0180	rings.cpp:63	ERROR	-
PDN	0181	domain.cpp:267	ERROR	-
PDN	0182	PdnGen.cc:472	WARN	-
PDN	0183	PdnGen.cc:301	WARN	-
PDN	0184	PdnGen.cc:322	ERROR	-
PDN	0185	[straps.cpp:127](“{})	grid	-
“with	total	strap	ERROR	-
PDN	0186	grid.cpp:116	ERROR	-
PDN	0187	straps.cpp:82	ERROR	-
PDN	0188	grid.cpp:1676	ERROR	-
PDN	0189	PdnGen.cc:857	WARN	-
PDN	0190	straps.cpp:347	ERROR	-
PDN	0191	techlayer.cpp:147	ERROR	-
PDN	0192	grid.cpp:606	ERROR	-
PDN	0193	grid.cpp:614	ERROR	-
PDN	0194	grid.cpp:638	ERROR	-
PDN	0195	via.cpp:2821	WARN	-
PDN	0196	PdnGen.cc:355	ERROR	-
PDN	0197	power_cells.cpp:154	ERROR	-
PDN	0198	power_cells.cpp:124	ERROR	-
PDN	0199	pdn.tcl:153	ERROR	-
PDN	0220	power_cells.cpp:247	ERROR	-
PDN	0221	power_cells.cpp:300	ERROR	-
PDN	0222	power_cells.cpp:236	WARN	-
PDN	0223	power_cells.cpp:328	WARN	-
PDN	0224	grid_component.cpp:472	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PDN	0225	pdn.tcl:340	ERROR	-
PDN	0226	via_repair.cpp:130	WARN	-
PDN	0227	via.cpp:860	WARN	-
PDN	0228	PdnGen.cc:740	WARN	-
PDN	0229	PdnGen.cc:318	ERROR	-
PDN	0230	pdn.tcl:463	ERROR	-
PDN	0231	grid.cpp:1513	WARN	-
PDN	0232	PdnGen.cc:154	WARN	-
PDN	0233	PdnGen.cc:161	ERROR	-
PDN	0234	PdnGen.cc:874	WARN	-
PDN	0235	PdnGen.cc:879	ERROR	-
PDN	1001	pdn.tcl:99	ERROR	-
PDN	1002	pdn.tcl:103	ERROR	-
PDN	1003	pdn.tcl:108	ERROR	-
PDN	1004	pdn.tcl:112	ERROR	-
PDN	1005	pdn.tcl:121	ERROR	-
PDN	1006	pdn.tcl:136	ERROR	-
PDN	1007	pdn.tcl:290	ERROR	-
PDN	1008	pdn.tcl:295	ERROR	-
PDN	1009	pdn.tcl:301	ERROR	-
PDN	1010	pdn.tcl:354	ERROR	-
PDN	1011	pdn.tcl:416	ERROR	-
PDN	1012	pdn.tcl:420	ERROR	-
PDN	1013	pdn.tcl:424	ERROR	-
PDN	1014	pdn.tcl:430	ERROR	-
PDN	1015	pdn.tcl:436	ERROR	-
PDN	1016	pdn.tcl:438	ERROR	-
PDN	1017	pdn.tcl:450	ERROR	-
PDN	1019	pdn.tcl:554	ERROR	-
PDN	1020	pdn.tcl:556	ERROR	-
PDN	1021	pdn.tcl:592	ERROR	-
PDN	1022	pdn.tcl:812	ERROR	-
PDN	1023	pdn.tcl:826	ERROR	-
PDN	1024	pdn.tcl:835	WARN	-
PDN	1025	pdn.tcl:856	ERROR	-
PDN	1026	pdn.tcl:947	ERROR	-
PDN	1027	pdn.tcl:963	ERROR	-
PDN	1028	pdn.tcl:965	ERROR	-
PDN	1029	pdn.tcl:974	ERROR	-
PDN	1030	pdn.tcl:1018	ERROR	-
PDN	1032	pdn.tcl:1069	ERROR	-
PDN	1033	pdn.tcl:1116	ERROR	-
PDN	1034	pdn.tcl:1129	ERROR	-
PDN	1035	pdn.tcl:1148	ERROR	-
PDN	1036	pdn.tcl:1171	ERROR	-
PDN	1037	pdn.tcl:49	ERROR	-
PDN	1038	pdn.tcl:56	ERROR	-
PDN	1039	pdn.tcl:63	ERROR	-
PDN	1040	pdn.tcl:1466	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PDN	1041	pdn.tcl:1469	ERROR	-
PDN	1042	pdn.tcl:160	WARN	-
PDN	1043	pdn.tcl:860	ERROR	-
PDN	1044	pdn.tcl:977	ERROR	-
PDN	1045	pdn.tcl:890	ERROR	-
PDN	1046	pdn.tcl:224	ERROR	-
PDN	1047	pdn.tcl:1155	ERROR	-
PDN	1048	pdn.tcl:886	ERROR	-
PDN	1049	pdn.tcl:894	ERROR	-
PDN	1183	pdn.tcl:220	ERROR	-
PDN	1184	pdn.tcl:229	ERROR	-
PDN	1186	pdn.tcl:240	ERROR	-
PDN	1187	pdn.tcl:246	ERROR	-
PDN	1188	pdn.tcl:252	ERROR	-
PDN	1190	pdn.tcl:767	ERROR	-
PDN	1191	pdn.tcl:757	ERROR	-
PDN	1192	pdn.tcl:760	ERROR	-
PDN	9002	PdnGen.tcl:3142	WARN	-
PDN	9003	PdnGen.tcl:3146	WARN	-
PDN	9004	PdnGen.tcl:3156	WARN	-
PDN	9006	PdnGen.tcl:5360	WARN	-
PDN	9008	PdnGen.tcl:4928	INFO	-
PDN	9009	PdnGen.tcl:4975	INFO	-
PDN	9010	PdnGen.tcl:6783	INFO	-
PDN	9011	PdnGen.tcl:6819	INFO	-
PDN	9012	PdnGen.tcl:6829	INFO	-
PDN	9013	PdnGen.tcl:6833	INFO	-
PDN	9014	PdnGen.tcl:6835	INFO	-
PDN	9015	PdnGen.tcl:6864	INFO	-
PDN	9016	PdnGen.tcl:6884	INFO	-
PDN	9017	PdnGen.tcl:4230	ERROR	-
PDN	9018	PdnGen.tcl:6816	WARN	-
PDN	9019	PdnGen.tcl:1468	ERROR	-
PDN	9020	PdnGen.tcl:1640	ERROR	-
PDN	9021	PdnGen.tcl:1705	ERROR	-
PDN	9022	PdnGen.tcl:2918	ERROR	-
PDN	9023	PdnGen.tcl:2919	ERROR	-
PDN	9024	PdnGen.tcl:2926	ERROR	-
PDN	9025	PdnGen.tcl:3312	ERROR	-
PDN	9026	PdnGen.tcl:3458	ERROR	-
PDN	9027	PdnGen.tcl:4738	ERROR	-
PDN	9028	PdnGen.tcl:6894	ERROR	-
PDN	9029	PdnGen.tcl:4881	ERROR	-
PDN	9030	PdnGen.tcl:5020	ERROR	-
PDN	9032	PdnGen.tcl:5640	INFO	-
PDN	9033	PdnGen.tcl:1376	ERROR	-
PDN	9034	PdnGen.tcl:6786	INFO	-
PDN	9035	PdnGen.tcl:5720	WARN	-
PDN	9036	PdnGen.tcl:2752	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PDN	9037	PdnGen.tcl:4782	ERROR	-
PDN	9038	PdnGen.tcl:2725	WARN	-
PDN	9039	PdnGen.tcl:2746	WARN	-
PDN	9040	PdnGen.tcl:3175	WARN	-
PDN	9041	PdnGen.tcl:3181	WARN	-
PDN	9042	PdnGen.tcl:3189	WARN	-
PDN	9043	PdnGen.tcl:3195	WARN	-
PDN	9044	PdnGen.tcl:3076	ERROR	-
PDN	9045	PdnGen.tcl:3097	ERROR	-
PDN	9048	PdnGen.tcl:3979	ERROR	-
PDN	9051	PdnGen.tcl:5966	ERROR	-
PDN	9052	PdnGen.tcl:3038	ERROR	-
PDN	9055	PdnGen.tcl:5085	WARN	-
PDN	9056	PdnGen.tcl:5089	WARN	-
PDN	9062	PdnGen.tcl:6890	ERROR	-
PDN	9063	PdnGen.tcl:2905	WARN	-
PDN	9064	PdnGen.tcl:4039	WARN	-
PDN	9065	PdnGen.tcl:4042	WARN	-
PDN	9066	PdnGen.tcl:4045	WARN	-
PDN	9067	PdnGen.tcl:4048	WARN	-
PDN	9068	PdnGen.tcl:4051	ERROR	-
PDN	9069	PdnGen.tcl:4599	ERROR	-
PDN	9070	PdnGen.tcl:4644	ERROR	-
PDN	9071	PdnGen.tcl:4650	ERROR	-
PDN	9072	PdnGen.tcl:86	ERROR	-
PDN	9074	PdnGen.tcl:101	ERROR	-
PDN	9075	PdnGen.tcl:114	ERROR	-
PDN	9076	PdnGen.tcl:117	ERROR	-
PDN	9077	PdnGen.tcl:132	ERROR	-
PDN	9078	PdnGen.tcl:135	ERROR	-
PDN	9079	PdnGen.tcl:147	ERROR	-
PDN	9081	PdnGen.tcl:154	ERROR	-
PDN	9083	PdnGen.tcl:173	ERROR	-
PDN	9084	PdnGen.tcl:180	ERROR	-
PDN	9085	PdnGen.tcl:194	ERROR	-
PDN	9086	PdnGen.tcl:197	ERROR	-
PDN	9087	PdnGen.tcl:206	ERROR	-
PDN	9088	PdnGen.tcl:545	ERROR	-
PDN	9090	PdnGen.tcl:1191	ERROR	-
PDN	9095	PdnGen.tcl:273	ERROR	-
PDN	9109	PdnGen.tcl:229	ERROR	-
PDN	9110	PdnGen.tcl:284	ERROR	-
PDN	9111	PdnGen.tcl:296	ERROR	-
PDN	9112	PdnGen.tcl:306	WARN	-
PDN	9114	PdnGen.tcl:702	ERROR	-
PDN	9115	PdnGen.tcl:907	ERROR	-
PDN	9116	PdnGen.tcl:920	ERROR	-
PDN	9117	PdnGen.tcl:933	ERROR	-
PDN	9118	PdnGen.tcl:946	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PDN	9119	PdnGen.tcl:973	ERROR	-
PDN	9121	PdnGen.tcl:237	ERROR	-
PDN	9124	PdnGen.tcl:736	ERROR	-
PDN	9125	PdnGen.tcl:959	ERROR	-
PDN	9126	PdnGen.tcl:1004	ERROR	-
PDN	9127	PdnGen.tcl:317	ERROR	-
PDN	9128	PdnGen.tcl:358	ERROR	-
PDN	9129	PdnGen.tcl:395	ERROR	-
PDN	9130	PdnGen.tcl:442	ERROR	-
PDN	9138	PdnGen.tcl:452	ERROR	-
PDN	9139	PdnGen.tcl:257	ERROR	-
PDN	9140	PdnGen.tcl:263	ERROR	-
PDN	9141	PdnGen.tcl:265	ERROR	-
PDN	9146	PdnGen.tcl:253	ERROR	-
PDN	9147	PdnGen.tcl:1139	ERROR	-
PDN	9148	PdnGen.tcl:1142	ERROR	-
PDN	9149	PdnGen.tcl:831	ERROR	-
PDN	9150	PdnGen.tcl:845	WARN	-
PDN	9151	PdnGen.tcl:864	ERROR	-
PDN	9152	PdnGen.tcl:878	WARN	-
PDN	9153	PdnGen.tcl:1151	ERROR	-
PDN	9154	PdnGen.tcl:1154	ERROR	-
PDN	9155	PdnGen.tcl:1164	ERROR	-
PDN	9156	PdnGen.tcl:1167	ERROR	-
PDN	9158	PdnGen.tcl:753	ERROR	-
PDN	9159	PdnGen.tcl:764	ERROR	-
PDN	9160	PdnGen.tcl:5687	ERROR	-
PDN	9164	PdnGen.tcl:496	ERROR	-
PDN	9165	PdnGen.tcl:1093	ERROR	-
PDN	9166	PdnGen.tcl:1277	ERROR	-
PDN	9168	PdnGen.tcl:2994	ERROR	-
PDN	9169	PdnGen.tcl:6132	WARN	-
PDN	9170	PdnGen.tcl:6152	WARN	-
PDN	9171	PdnGen.tcl:6160	WARN	-
PDN	9172	PdnGen.tcl:6169	WARN	-
PDN	9176	PdnGen.tcl:377	ERROR	-
PDN	9177	PdnGen.tcl:4788	WARN	-
PDN	9178	PdnGen.tcl:486	ERROR	-
PDN	9179	PdnGen.tcl:476	ERROR	-
PDN	9180	PdnGen.tcl:342	ERROR	-
PDN	9181	PdnGen.tcl:326	WARN	-
PDN	9190	PdnGen.tcl:417	ERROR	-
PDN	9191	PdnGen.tcl:3804	ERROR	-
PDN	9192	PdnGen.tcl:3809	ERROR	-
PDN	9193	PdnGen.tcl:3822	ERROR	-
PDN	9194	PdnGen.tcl:505	ERROR	-
PDN	9195	PdnGen.tcl:517	ERROR	-
PDN	9196	PdnGen.tcl:3859	ERROR	-
PDN	9197	PdnGen.tcl:3850	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PDN	9248	PdnGen.tcl:5654	ERROR	-
PPL	0001	IOPlacer.cpp:1311	INFO	-
PPL	0002	IOPlacer.cpp:1316	INFO	-
PPL	0003	IOPlacer.cpp:1318	INFO	-
PPL	0004	IOPlacer.cpp:1322	INFO	-
PPL	0005	IOPlacer.cpp:1324	INFO	-
PPL	0006	IOPlacer.cpp:1325	INFO	-
PPL	0007	IOPlacer.cpp:1998	INFO	-
PPL	0008	IOPlacer.cpp:1258	INFO	-
PPL	0009	IOPlacer.cpp:1261	INFO	-
PPL	0010	IOPlacer.cpp:1336	INFO	-
PPL	0011	IOPlacer.cpp:880	ERROR	-
PPL	0012	IOPlacer.cpp:2201	INFO	-
PPL	0013	IOPlacer.cpp:1249	ERROR	-
PPL	0015	IOPlacer.tcl:466	WARN	-
PPL	0016	IOPlacer.tcl:165	ERROR	-
PPL	0017	IOPlacer.tcl:484	ERROR	-
PPL	0018	IOPlacer.tcl:490	ERROR	-
PPL	0019	IOPlacer.tcl:519	ERROR	-
PPL	0020	IOPlacer.cpp:1403	ERROR	-
PPL	0021	IOPlacer.tcl:527	ERROR	-
PPL	0023	IOPlacer.tcl:545	ERROR	-
PPL	0024	IOPlacer.cpp:871	ERROR	-
PPL	0025	IOPlacer.tcl:585	ERROR	-
PPL	0026	IOPlacer.tcl:588	ERROR	-
PPL	0027	IOPlacer.tcl:627	ERROR	-
PPL	0028	IOPlacer.tcl:640	ERROR	-
PPL	0029	IOPlacer.tcl:683	ERROR	-
PPL	0030	IOPlacer.tcl:691	ERROR	-
PPL	0031	IOPlacer.tcl:451	ERROR	-
PPL	0032	IOPlacer.tcl:456	ERROR	-
PPL	0033	HungarianMatching.cpp:124	WARN	-
PPL	0034	IOPlacer.cpp:2251	ERROR	-
PPL	0035	IOPlacer.cpp:692	ERROR	-
PPL	0036	IOPlacer.cpp:1343	WARN	-
PPL	0037	IOPlacer.cpp:1352	WARN	-
PPL	0038	IOPlacer.cpp:2671	WARN	-
PPL	0039	IOPlacer.cpp:2065	ERROR	-
PPL	0040	IOPlacer.cpp:912	ERROR	-
PPL	0041	IOPlacer.tcl:596	INFO	-
PPL	0042	IOPlacer.cpp:1194	WARN	-
PPL	0043	IOPlacer.tcl:603	WARN	-
PPL	0044	IOPlacer.cpp:1917	INFO	-
PPL	0045	IOPlacer.tcl:531	ERROR	-
PPL	0046	IOPlacer.tcl:549	ERROR	-
PPL	0047	IOPlacer.tcl:219	WARN	-
PPL	0048	IOPlacer.cpp:1591	INFO	-
PPL	0049	IOPlacer.tcl:171	INFO	-
PPL	0050	IOPlacer.tcl:708	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PPL	0051	IOPlacer.tcl:713	ERROR	-
PPL	0052	IOPlacer.tcl:55	ERROR	-
PPL	0053	IOPlacer.tcl:58	ERROR	-
PPL	0054	IOPlacer.tcl:65	ERROR	-
PPL	0055	IOPlacer.tcl:87	ERROR	-
PPL	0056	IOPlacer.tcl:93	ERROR	-
PPL	0057	IOPlacer.tcl:99	ERROR	-
PPL	0058	IOPlacer.tcl:208	ERROR	-
PPL	0059	IOPlacer.tcl:192	ERROR	-
PPL	0060	IOPlacer.tcl:732	INFO	-
PPL	0061	IOPlacer.tcl:745	ERROR	-
PPL	0062	IOPlacer.cpp:1313	INFO	-
PPL	0063	IOPlacer.tcl:83	ERROR	-
PPL	0064	IOPlacer.tcl:374	ERROR	-
PPL	0065	IOPlacer.tcl:380	ERROR	-
PPL	0066	IOPlacer.tcl:386	ERROR	-
PPL	0068	IOPlacer.tcl:390	ERROR	-
PPL	0069	IOPlacer.tcl:403	ERROR	-
PPL	0070	IOPlacer.cpp:2333	INFO	-
PPL	0071	IOPlacer.tcl:411	ERROR	-
PPL	0072	IOPlacer.cpp:236	ERROR	-
PPL	0073	IOPlacer.tcl:200	WARN	-
PPL	0075	IOPlacer.cpp:1699	WARN	-
PPL	0076	IOPlacer.cpp:1750	ERROR	-
PPL	0077	IOPlacer.cpp:1427	ERROR	-
PPL	0078	IOPlacer.cpp:1179	WARN	-
PPL	0079	IOPlacer.cpp:1498	ERROR	-
PPL	0080	IOPlacer.tcl:238	INFO	-
PPL	0081	IOPlacer.tcl:234	ERROR	-
PPL	0082	HungarianMatching.cpp:173	ERROR	-
PPL	0083	IOPlacer.tcl:135	ERROR	-
PPL	0084	IOPlacer.cpp:1683	WARN	-
PPL	0085	IOPlacer.cpp:315	ERROR	-
PPL	0086	HungarianMatching.cpp:326	ERROR	-
PPL	0087	IOPlacer.tcl:139	ERROR	-
PPL	0088	IOPlacer.cpp:2031	ERROR	-
PPL	0089	HungarianMatching.cpp:266	ERROR	-
PPL	0090	IOPlacer.cpp:357	ERROR	-
PPL	0091	IOPlacer.cpp:451	ERROR	-
PPL	0092	IOPlacer.cpp:2007	WARN	-
PPL	0093	IOPlacer.cpp:383	ERROR	-
PPL	0094	IOPlacer.cpp:2727	ERROR	-
PPL	0095	IOPlacer.tcl:228	ERROR	-
PPL	0096	IOPlacer.cpp:228	WARN	-
PPL	0097	IOPlacer.cpp:529	WARN	-
PPL	0098	IOPlacer.cpp:1814	ERROR	-
PPL	0099	IOPlacer.tcl:728	ERROR	-
PPL	0100	IOPlacer.cpp:565	INFO	-
PPL	0101	IOPlacer.cpp:476	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PPL	0104	IOPlacer.cpp:1843	ERROR	-
PPL	0106	IOPlacer.cpp:2177	WARN	-
PPL	0107	IOPlacer.cpp:2193	ERROR	-
PPL	0108	IOPlacer.tcl:353	ERROR	-
PPL	0109	IOPlacer.cpp:408	ERROR	-
PPL	0110	IOPlacer.cpp:1742	WARN	-
PPL	0111	IOPlacer.cpp:1766	ERROR	-
PPL	0112	SimulatedAnnealing.cpp:854	ERROR	-
PSM	0001	pdnsim.cpp:89	INFO	-
PSM	0004	ir_solver.cpp:1877	INFO	-
PSM	0008	ir_solver.cpp:167	WARN	-
PSM	0010	ir_solver.cpp:190	ERROR	-
PSM	0012	ir_solver.cpp:201	ERROR	-
PSM	0014	ir_solver.cpp:382	ERROR	-
PSM	0015	ir_solver.cpp:437	INFO	-
PSM	0016	ir_solver.cpp:472	WARN	-
PSM	0017	ir_solver.cpp:487	WARN	-
PSM	0018	ir_solver.cpp:495	WARN	-
PSM	0022	ir_solver.cpp:414	INFO	-
PSM	0024	ir_solver.cpp:728	WARN	-
PSM	0028	pdnsim.tcl:199	ERROR	-
PSM	0030	ir_solver.cpp:1268	WARN	-
PSM	0031	ir_solver.cpp:1372	INFO	-
PSM	0032	ir_solver.cpp:1145	WARN	-
PSM	0033	ir_solver.cpp:1156	WARN	-
PSM	0035	ir_solver.cpp:1125	ERROR	-
PSM	0036	ir_solver.cpp:1178	ERROR	-
PSM	0037	ir_solver.cpp:1199	ERROR	-
PSM	0038	ir_solver.cpp:1446	WARN	-
PSM	0039	ir_solver.cpp:1456	WARN	-
PSM	0040	ir_solver.cpp:1478	INFO	-
PSM	0041	ir_solver.cpp:1791	ERROR	-
PSM	0042	ir_solver.cpp:711	ERROR	-
PSM	0045	gmat.cpp:191	ERROR	-
PSM	0046	gmat.cpp:204	ERROR	-
PSM	0047	gmat.cpp:206	ERROR	-
PSM	0048	gmat.cpp:289	INFO	-
PSM	0049	gmat.cpp:341	ERROR	-
PSM	0050	gmat.cpp:395	WARN	-
PSM	0051	gmat.cpp:570	ERROR	-
PSM	0052	gmat.cpp:596	ERROR	-
PSM	0054	pdnsim.tcl:59	ERROR	-
PSM	0055	pdnsim.tcl:97	ERROR	-
PSM	0056	pdnsim.tcl:103	ERROR	-
PSM	0057	pdnsim.tcl:117	ERROR	-
PSM	0058	pdnsim.tcl:129	ERROR	-
PSM	0059	pdnsim.tcl:150	ERROR	-
PSM	0060	pdnsim.tcl:159	ERROR	-
PSM	0061	pdnsim.tcl:175	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
PSM	0062	pdnsim.tcl:191	ERROR	-
PSM	0063	ir_solver.cpp:505	WARN	-
PSM	0064	ir_solver.cpp:384	INFO	-
PSM	0065	ir_solver.cpp:528	WARN	-
PSM	0066	ir_solver.cpp:1222	ERROR	-
PSM	0067	ir_solver.cpp:1288	WARN	-
PSM	0068	pdnsim.cpp:255	ERROR	-
PSM	0069	pdnsim.tcl:125	ERROR	-
PSM	0070	ir_solver.cpp:1360	WARN	-
PSM	0071	ir_solver.cpp:664	WARN	-
PSM	0073	ir_solver.cpp:1318	INFO	-
PSM	0075	ir_solver.cpp:462	ERROR	-
PSM	0076	ir_solver.cpp:1339	INFO	-
PSM	0077	pdnsim.tcl:73	ERROR	-
PSM	0078	pdnsim.cpp:167	ERROR	-
PSM	0079	pdnsim.cpp:286	ERROR	-
PSM	0080	gmat.cpp:111	WARN	-
PSM	0081	ir_solver.cpp:873	ERROR	-
PSM	0082	ir_solver.cpp:1333	ERROR	-
PSM	0084	pdnsim.cpp:101	ERROR	-
PSM	0085	pdnsim.tcl:154	ERROR	-
PSM	0086	ir_solver.cpp:139	ERROR	-
PSM	0087	ir_solver.cpp:129	ERROR	-
PSM	0088	ir_solver.cpp:126	ERROR	-
PSM	0089	ir_solver.cpp:434	ERROR	-
PSM	0090	ir_solver.cpp:292	ERROR	-
PSM	0091	ir_solver.cpp:327	ERROR	-
PSM	0092	ir_solver.cpp:1705	ERROR	-
PSM	0093	ir_solver.cpp:405	ERROR	-
PSM	0094	ir_solver.cpp:1471	WARN	-
RCX	0001	ext.cpp:312	INFO	-
RCX	0002	ext.cpp:320	ERROR	-
RCX	0003	netRC.cpp:2359	WARN	-
RCX	0004	netRC.cpp:2355	WARN	-
RCX	0005	netRC.cpp:2371	WARN	-
RCX	0007	extBench.cpp:449	INFO	-
RCX	0008	ext.cpp:233	INFO	-
RCX	0015	ext.cpp:250	INFO	-
RCX	0016	ext.cpp:283	INFO	-
RCX	0017	ext.cpp:306	INFO	-
RCX	0019	ext.cpp:369	INFO	-
RCX	0021	ext.cpp:422	INFO	-
RCX	0029	ext.cpp:187	INFO	-
RCX	0030	ext.cpp:198	ERROR	-
RCX	0031	ext.cpp:207	INFO	-
RCX	0040	netRC.cpp:1905	INFO	-
RCX	0042	extSpef.cpp:1853	INFO	-
RCX	0043	extFlow.cpp:1243	INFO	-
RCX	0044	extSpefIn.cpp:2797	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
RCX	0045	netRC.cpp:2008	INFO	-
RCX	0047	extSpef.cpp:1898	INFO	-
RCX	0048	extSpefIn.cpp:2788	WARN	-
RCX	0049	extSpefIn.cpp:2791	WARN	-
RCX	0050	extSpefIn.cpp:2794	WARN	-
RCX	0052	extSpefIn.cpp:2801	ERROR	-
RCX	0055	extBench.cpp:281	INFO	-
RCX	0057	extBench.cpp:334	INFO	-
RCX	0058	extBench.cpp:389	INFO	-
RCX	0060	extSpefIn.cpp:2735	INFO	-
RCX	0065	extmain.cpp:771	INFO	-
RCX	0069	extRCmodel.cpp:3447	INFO	-
RCX	0072	extRCmodel.cpp:3565	INFO	-
RCX	0074	extmeasure.cpp:66	WARN	-
RCX	0076	extSpefIn.cpp:584	WARN	-
RCX	0077	extSpefIn.cpp:813	WARN	-
RCX	0078	extSpefIn.cpp:2351	WARN	-
RCX	0079	extmeasure.cpp:1843	INFO	-
RCX	0081	extFlow.cpp:294	ERROR	-
RCX	0082	extFlow.cpp:336	ERROR	-
RCX	0107	netRC.cpp:2016	WARN	-
RCX	0108	netRC.cpp:469	INFO	-
RCX	0109	netRC.cpp:483	INFO	-
RCX	0110	netRC.cpp:631	INFO	-
RCX	0111	netRC.cpp:686	WARN	-
RCX	0112	netRC.cpp:785	ERROR	-
RCX	0113	netRC.cpp:790	ERROR	-
RCX	0114	netRC.cpp:798	WARN	-
RCX	0115	netRC.cpp:805	ERROR	-
RCX	0120	netRC.cpp:1237	WARN	-
RCX	0121	netRC.cpp:1384	INFO	-
RCX	0122	netRC.cpp:1400	INFO	-
RCX	0127	netRC.cpp:1808	WARN	-
RCX	0128	netRC.cpp:1830	INFO	-
RCX	0129	netRC.cpp:1834	WARN	-
RCX	0134	netRC.cpp:2250	INFO	-
RCX	0135	netRC.cpp:2176	WARN	-
RCX	0136	netRC.cpp:2200	INFO	-
RCX	0137	netRC.cpp:2270	INFO	-
RCX	0138	extmain.cpp:452	WARN	-
RCX	0139	extmain.cpp:444	WARN	-
RCX	0140	extmain.cpp:664	INFO	-
RCX	0141	extmain.cpp:749	INFO	-
RCX	0142	extmain.cpp:761	INFO	-
RCX	0143	extmain.cpp:773	INFO	-
RCX	0147	ext.cpp:172	ERROR	-
RCX	0148	ext.cpp:226	WARN	-
RCX	0149	OpenRCX.tcl:330	WARN	-
RCX	0152	extprocess.cpp:294	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
RCX	0153	extprocess.cpp:325	WARN	-
RCX	0154	extprocess.cpp:530	WARN	-
RCX	0158	extprocess.cpp:269	WARN	-
RCX	0159	extprocess.cpp:547	ERROR	-
RCX	0171	extSpef.cpp:306	ERROR	-
RCX	0172	extSpef.cpp:311	ERROR	-
RCX	0175	extSpef.cpp:1360	WARN	-
RCX	0176	extSpef.cpp:1680	INFO	-
RCX	0178	extSpefIn.cpp:2434	INFO	-
RCX	0208	extRCmodel.cpp:148	INFO	-
RCX	0216	extRCmodel.cpp:3577	INFO	-
RCX	0217	extRCmodel.cpp:3583	INFO	-
RCX	0218	extRCmodel.cpp:3632	INFO	-
RCX	0219	extRCmodel.cpp:3645	INFO	-
RCX	0220	extRCmodel.cpp:3668	INFO	-
RCX	0221	extRCmodel.cpp:3681	INFO	-
RCX	0222	extRCmodel.cpp:3927	WARN	-
RCX	0223	extRCmodel.cpp:3951	WARN	-
RCX	0239	extFlow.cpp:1394	WARN	-
RCX	0240	extFlow.cpp:1405	WARN	-
RCX	0252	extmain.cpp:54	ERROR	-
RCX	0258	extSpefIn.cpp:144	WARN	-
RCX	0259	extSpefIn.cpp:181	ERROR	-
RCX	0260	extSpefIn.cpp:461	ERROR	-
RCX	0261	extSpefIn.cpp:639	WARN	-
RCX	0262	extSpefIn.cpp:555	ERROR	-
RCX	0263	extSpefIn.cpp:612	ERROR	-
RCX	0264	extSpefIn.cpp:957	WARN	-
RCX	0265	extSpefIn.cpp:1155	INFO	-
RCX	0266	extSpefIn.cpp:1218	INFO	-
RCX	0267	extSpefIn.cpp:1420	INFO	-
RCX	0269	extSpefIn.cpp:1584	WARN	-
RCX	0270	extSpefIn.cpp:1628	WARN	-
RCX	0271	extSpefIn.cpp:1707	WARN	-
RCX	0272	extSpefIn.cpp:1724	WARN	-
RCX	0273	extSpefIn.cpp:1767	ERROR	-
RCX	0274	extSpefIn.cpp:1774	WARN	-
RCX	0275	extSpefIn.cpp:1782	WARN	-
RCX	0276	extSpefIn.cpp:1784	WARN	-
RCX	0277	extSpefIn.cpp:1828	WARN	-
RCX	0278	extSpefIn.cpp:1855	WARN	-
RCX	0279	extSpefIn.cpp:1882	WARN	-
RCX	0280	extSpefIn.cpp:1940	WARN	-
RCX	0281	extSpefIn.cpp:1996	WARN	-
RCX	0282	extSpefIn.cpp:2081	WARN	-
RCX	0283	extSpefIn.cpp:2340	INFO	-
RCX	0284	extSpefIn.cpp:2348	WARN	-
RCX	0285	extSpefIn.cpp:2686	WARN	-
RCX	0286	extSpefIn.cpp:2480	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
RCX	0287	extSpefIn.cpp:2500	INFO	-
RCX	0288	extSpefIn.cpp:2507	INFO	-
RCX	0289	extSpefIn.cpp:2521	INFO	-
RCX	0290	extSpefIn.cpp:2531	INFO	-
RCX	0291	extSpefIn.cpp:2571	INFO	-
RCX	0292	extSpefIn.cpp:2683	WARN	-
RCX	0293	extSpefIn.cpp:2815	INFO	-
RCX	0294	extSpefIn.cpp:2832	INFO	-
RCX	0295	extSpefIn.cpp:2940	WARN	-
RCX	0296	extSpefIn.cpp:2970	WARN	-
RCX	0297	extSpefIn.cpp:2975	WARN	-
RCX	0298	extSpefIn.cpp:2977	WARN	-
RCX	0357	OpenRCX.tcl:233	ERROR	-
RCX	0358	extRCmodel.cpp:3571	INFO	-
RCX	0374	extSpefIn.cpp:1691	WARN	-
RCX	0376	netRC.cpp:2421	INFO	-
RCX	0378	ext.cpp:176	ERROR	-
RCX	0380	ext.cpp:366	ERROR	-
RCX	0381	ext.cpp:417	ERROR	-
RCX	0404	extSpefIn.cpp:2491	INFO	-
RCX	0405	extSpefIn.cpp:1874	WARN	-
RCX	0406	extSpefIn.cpp:1821	WARN	-
RCX	0407	extSpefIn.cpp:1863	WARN	-
RCX	0410	extRCmodel.cpp:3619	INFO	-
RCX	0411	extRCmodel.cpp:2772	INFO	-
RCX	0414	extRCmodel.cpp:2766	INFO	-
RCX	0416	extRCmodel.cpp:2748	INFO	-
RCX	0417	extRCmodel.cpp:2730	INFO	-
RCX	0418	extRCmodel.cpp:2706	WARN	-
RCX	0431	netRC.cpp:1341	INFO	-
RCX	0433	netRC.cpp:1317	INFO	-
RCX	0434	netRC.cpp:1347	INFO	-
RCX	0435	netRC.cpp:1686	INFO	-
RCX	0436	netRC.cpp:1874	INFO	-
RCX	0437	netRC.cpp:78	INFO	-
RCX	0438	netRC.cpp:67	INFO	-
RCX	0439	netRC.cpp:1909	INFO	-
RCX	0440	netRC.cpp:1920	INFO	-
RCX	0442	extFlow.cpp:1364	INFO	-
RCX	0443	extSpef.cpp:1859	INFO	-
RCX	0444	extSpefIn.cpp:2361	INFO	-
RCX	0445	extSpefIn.cpp:2653	INFO	-
RCX	0447	extSpefIn.cpp:2776	WARN	-
RCX	0448	extSpefIn.cpp:2760	WARN	-
RCX	0449	extSpefIn.cpp:2822	INFO	-
RCX	0451	extSpefIn.cpp:2825	INFO	-
RCX	0452	extSpefIn.cpp:71	WARN	-
RCX	0456	extmeasure.cpp:947	INFO	-
RCX	0458	extmeasure.cpp:1016	INFO	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
RCX	0459	extmeasure.cpp:626	INFO	-
RCX	0460	extmeasure.cpp:76	WARN	-
RCX	0463	extSpefIn.cpp:2725	INFO	-
RCX	0464	extSpefIn.cpp:2738	INFO	-
RCX	0465	extSpef.cpp:1895	INFO	-
RCX	0468	netRC.cpp:1691	ERROR	-
RCX	0472	netRC.cpp:1367	INFO	-
RCX	0474	netRC.cpp:2138	INFO	-
RCX	0475	netRC.cpp:2234	INFO	-
RCX	0476	extmain.cpp:763	INFO	-
RCX	0480	netRC.cpp:2439	INFO	-
RCX	0485	extRCmodel.cpp:2642	INFO	-
RCX	0487	netRC.cpp:1713	ERROR	-
RCX	0489	extRCmodel.cpp:3154	ERROR	-
RCX	0490	extRCmodel.cpp:3449	ERROR	-
RCX	0491	extRCmodel.cpp:3458	ERROR	-
RCX	0497	extmain.cpp:484	ERROR	-
RMP	0001	blif.cpp:105	ERROR	-
RMP	0002	blif.cpp:357	INFO	-
RMP	0003	blif.cpp:381	ERROR	-
RMP	0004	blif.cpp:404	ERROR	-
RMP	0005	blif.cpp:426	INFO	-
RMP	0006	blif.cpp:430	INFO	-
RMP	0007	blif.cpp:455	INFO	-
RMP	0008	blif.cpp:473	INFO	-
RMP	0009	blif.cpp:516	INFO	-
RMP	0010	blif.cpp:569	INFO	-
RMP	0012	rmp.tcl:90	ERROR	-
RMP	0016	Restructure.cpp:620	ERROR	-
RMP	0020	Restructure.cpp:485	ERROR	-
RMP	0021	Restructure.cpp:292	INFO	-
RMP	0025	Restructure.cpp:642	WARN	-
RMP	0026	Restructure.cpp:225	ERROR	-
RMP	0032	rmp.tcl:106	WARN	-
RMP	0033	rmp.tcl:122	WARN	-
RMP	0034	blif.cpp:418	ERROR	-
RMP	0035	Restructure.cpp:453	WARN	-
RMP	0036	Restructure.cpp:594	WARN	-
RMP	0037	Restructure.cpp:299	ERROR	-
RMP	0076	blif.cpp:536	ERROR	-
RMP	0146	blif.cpp:590	INFO	-
RSZ	0001	Resizer.tcl:145	ERROR	-
RSZ	0002	Resizer.tcl:150	ERROR	-
RSZ	0003	Resizer.tcl:237	ERROR	-
RSZ	0004	Resizer.tcl:586	ERROR	-
RSZ	0005	Resizer.tcl:234	ERROR	-
RSZ	0010	Resizer.tcl:198	WARN	-
RSZ	0011	Resizer.tcl:201	WARN	-
RSZ	0014	Resizer.tcl:608	WARN	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
RSZ	0020	Resizer.tcl:506	WARN	-
RSZ	0021	Resizer.tcl:553	WARN	-
RSZ	0022	Resizer.cc:377	ERROR	-
RSZ	0023	Resizer.cc:2782	ERROR	-
RSZ	0025	RepairSetup.cc:289	ERROR	-
RSZ	0026	Resizer.cc:272	INFO	-
RSZ	0027	Resizer.cc:428	INFO	-
RSZ	0028	Resizer.cc:549	INFO	-
RSZ	0030	RepairSetup.cc:313	INFO	-
RSZ	0031	RepairSetup.cc:316	INFO	-
RSZ	0032	RepairHold.cc:305	INFO	-
RSZ	0033	RepairHold.cc:316	INFO	-
RSZ	0034	RepairDesign.cc:123	INFO	-
RSZ	0035	RepairDesign.cc:126	INFO	-
RSZ	0036	RepairDesign.cc:129	INFO	-
RSZ	0037	RepairDesign.cc:132	INFO	-
RSZ	0038	RepairDesign.cc:135	INFO	-
RSZ	0039	RepairDesign.cc:140	INFO	-
RSZ	0040	RepairSetup.cc:268	INFO	-
RSZ	0041	RepairSetup.cc:276	INFO	-
RSZ	0042	Resizer.cc:1744	INFO	-
RSZ	0043	RepairSetup.cc:279	INFO	-
RSZ	0044	RepairSetup.cc:319	INFO	-
RSZ	0045	RepairSetup.cc:271	INFO	-
RSZ	0046	RepairHold.cc:263	INFO	-
RSZ	0047	RepairDesign.cc:269	INFO	-
RSZ	0048	RepairDesign.cc:272	INFO	-
RSZ	0049	RepairSetup.cc:282	INFO	-
RSZ	0050	RepairHold.cc:312	ERROR	-
RSZ	0051	RepairDesign.cc:319	INFO	-
RSZ	0052	RepairDesign.cc:322	INFO	-
RSZ	0053	RepairDesign.cc:325	INFO	-
RSZ	0054	RepairDesign.cc:328	INFO	-
RSZ	0055	RepairDesign.cc:331	INFO	-
RSZ	0056	RepairDesign.cc:337	INFO	-
RSZ	0057	RepairDesign.cc:340	INFO	-
RSZ	0058	Resizer.tcl:624	INFO	-
RSZ	0060	RepairHold.cc:309	ERROR	-
RSZ	0061	Resizer.tcl:163	INFO	-
RSZ	0062	RepairSetup.cc:286	WARN	-
RSZ	0064	RepairHold.cc:301	WARN	-
RSZ	0065	Resizer.tcl:620	WARN	-
RSZ	0066	RepairHold.cc:299	WARN	-
RSZ	0067	Resizer.tcl:564	WARN	-
RSZ	0068	Resizer.cc:1478	ERROR	-
RSZ	0069	SteinerTree.cc:92	WARN	-
RSZ	0070	Resizer.cc:2242	ERROR	-
RSZ	0071	Rebuffer.cc:297	CRITICAL	-
RSZ	0072	RepairDesign.cc:661	CRITICAL	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
RSZ	0073	BufferedNet.cc:660	WARN	-
RSZ	0074	BufferedNet.cc:664	WARN	-
RSZ	0075	Rebuffer.cc:132	WARN	-
RSZ	0076	Resizer.tcl:419	WARN	-
RSZ	0077	Resizer.cc:2995	WARN	-
RSZ	0078	BufferedNet.cc:83	CRITICAL	-
RSZ	0079	BufferedNet.cc:118	CRITICAL	-
RSZ	0080	BufferedNet.cc:146	CRITICAL	-
RSZ	0081	BufferedNet.cc:176	CRITICAL	-
RSZ	0082	BufferedNet.cc:346	CRITICAL	-
RSZ	0083	RepairDesign.cc:1062	CRITICAL	-
RSZ	0084	Resizer.cc:567	WARN	-
RSZ	0085	Resizer.cc:476	WARN	-
RSZ	0086	Resizer.cc:1353	ERROR	-
RSZ	0088	Resizer.cc:2199	WARN	-
RSZ	0089	Resizer.cc:2214	ERROR	-
RSZ	0090	PreChecks.cc:81	ERROR	-
RSZ	0092	SteinerTree.cc:394	ERROR	-
RSZ	0093	SteinerTree.cc:323	ERROR	-
RSZ	0094	RepairSetup.cc:138	INFO	-
RSZ	0095	Resizer.tcl:514	WARN	-
RSZ	0125	RecoverPower.cc:201	ERROR	-
RSZ	0141	RecoverPower.cc:198	INFO	-
RSZ	0142	RecoverPower.cc:176	INFO	-
RSZ	3111	RecoverPower.cc:222	INFO	-
STA	1000	dbSta.cc:524	ERROR	-
STT	0001	SteinerTreeBuilder.tcl:49	ERROR	-
STT	0002	SteinerTreeBuilder.tcl:72	ERROR	-
STT	0003	SteinerTreeBuilder.tcl:86	ERROR	-
STT	0004	SteinerTreeBuilder.cpp:248	ERROR	-
STT	0005	SteinerTreeBuilder.cpp:267	ERROR	-
STT	0006	SteinerTreeBuilder.tcl:102	ERROR	-
STT	0007	SteinerTreeBuilder.cpp:304	ERROR	-
TAP	0003	tapcell.cpp:463	INFO	-
TAP	0004	tapcell.cpp:466	INFO	-
TAP	0005	tapcell.cpp:164	INFO	-
TAP	0010	tapcell.tcl:168	ERROR	-
TAP	0011	tapcell.tcl:178	ERROR	-
TAP	0014	tapcell.tcl:73	WARN	-
TAP	0015	tapcell.tcl:142	WARN	-
TAP	0016	tapcell.tcl:146	WARN	-
TAP	0020	tapcell.cpp:1140	ERROR	-
TAP	0032	tapcell.cpp:266	WARN	-
TAP	0033	tapcell.cpp:301	ERROR	-
TAP	0034	tapcell.tcl:212	ERROR	-
TAP	0035	tapcell.i:62	ERROR	-
TAP	0100	tapcell.tcl:245	INFO	-
TAP	0101	tapcell.tcl:247	INFO	-
TAP	0102	tapcell.tcl:368	ERROR	-

continues on next page

Table 1 – continued from previous page

Tool	Code	Filename:Line Number	Type	Information
TAP	0103	tapcell.tcl:395	ERROR	-
TAP	0104	tapcell.cpp:1341	ERROR	-
UPF	0001	upf.cpp:50	WARN	-
UPF	0002	upf.cpp:66	WARN	-
UPF	0003	upf.cpp:84	WARN	-
UPF	0004	upf.cpp:97	WARN	-
UPF	0005	upf.cpp:108	WARN	-
UPF	0006	upf.cpp:124	WARN	-
UPF	0007	upf.cpp:144	WARN	-
UPF	0008	upf.cpp:163	WARN	-
UPF	0009	upf.cpp:182	WARN	-
UPF	0010	upf.cpp:201	WARN	-
UPF	0011	upf.cpp:220	WARN	-
UPF	0012	upf.cpp:244	WARN	-
UPF	0013	upf.cpp:255	WARN	-
UPF	0014	upf.cpp:300	WARN	-
UPF	0015	upf.cpp:311	WARN	-
UPF	0016	upf.cpp:331	WARN	-
UPF	0017	upf.cpp:401	WARN	-
UPF	0018	upf.cpp:420	WARN	-
UPF	0019	upf.cpp:445	ERROR	-
UPF	0020	upf.cpp:464	WARN	-
UPF	0021	upf.cpp:474	WARN	-
UPF	0022	upf.cpp:482	WARN	-
UPF	0023	upf.cpp:501	WARN	-
UPF	0024	upf.cpp:605	WARN	-
UPF	0025	upf.cpp:629	WARN	-
UPF	0026	upf.cpp:665	WARN	-
UPF	0027	upf.cpp:801	WARN	-
UPF	0028	upf.cpp:849	WARN	-
UPF	0029	upf.cpp:913	ERROR	-
UPF	0030	upf.cpp:950	WARN	-
UPF	0031	upf.cpp:978	WARN	-
UPF	0033	upf.tcl:51	ERROR	-
UPF	0034	upf.tcl:57	ERROR	-
UPF	0036	upf.tcl:325	ERROR	-
UPF	0037	upf.tcl:333	ERROR	-
UPF	0040	upf.tcl:388	ERROR	-
UTL	0001	CFileUtils.cpp:8	ERROR	-
UTL	0002	CFileUtils.cpp:25	ERROR	-
UTL	0003	CFileUtils.cpp:35	ERROR	-
UTL	0004	CFileUtils.cpp:56	ERROR	-
UTL	0005	ScopedTemporaryFile.cpp:12	ERROR	-
UTL	0006	ScopedTemporaryFile.cpp:15	INFO	-
UTL	0007	ScopedTemporaryFile.cpp:19	ERROR	-
UTL	0008	ScopedTemporaryFile.cpp:27	WARN	-
UTL	0009	ScopedTemporaryFile.cpp:31	WARN	-

5.3 Getting Involved

Thank you for taking the time to read this document and to contribute. The OpenROAD project will not reach all of its objectives without help!

Possible ways to contribute to the OpenROAD application:

- Tool improvements
- New tools
- Improvements to documentation, including this document
- Star our project and repos so we can see the number of people who are interested

5.3.1 Licensing Contributions

As much as possible, all contributions should be licensed using the BSD3 license. You can propose another license if you must, but contributions made with BSD3 fit best with the spirit of OpenROAD's permissive open-source philosophy. We do have exceptions in the project, but over time we hope that all contributions will be BSD3, or some other permissive license such as MIT or Apache2.0.

5.3.2 Contributing Scripts and Code

We follow the Google C++ style guide. If you find code in our project that does *not* follow this guide, then within each file that you edit, follow the style in that file.

Please pay careful attention to the *tool checklist* for all code. If you want to add or improve functionality in OpenROAD, please start with the top-level app repo. You can see in the `src` directory that submodules exist pointing to tested versions of the other relevant repos in the project. Please look at the tool workflow in the developer guide *document* to work with the app and its submodule repos in an efficient way.

Please run clang-format on all the C++ source files that you change, before committing. In the root directory of the OpenROAD repository there is the file `.clang-format` that defines all coding formatting rules.

Please pay attention to the test directory and be sure to add tests for any code changes that you make, using open-source PDK and design information. We provide the `nangate45` PDK in the OpenROAD-flow-scripts repo to help with this. Pull requests with code changes are unlikely to be accepted without accompanying test cases. There are many examples tests. Each repo has a test directory as well with tests you should run and add to if you modify something in one of the submodules.

For changes that claim to improve QoR or PPA, please run many tests and ensure that the improvement is not design-specific. There are designs in the OpenROAD-flow-scripts repo which can be used unless the improvement is technology-specific.

Do not add runtime or build dependencies without serious thought. For a project like OpenROAD with many application subcomponents, the software architecture can quickly get out of control. Changes with lots of new dependencies which are not necessary are less likely to be integrated.

If you want to add Tcl code to define a new tool command, look at pdngen as an example of how to do so. Take a look at the CMake file which automatically sources the Tcl code and the Tcl file itself.

To accept contributions, we require each commit to be made with a DCO (Developer Certificate of Origin) attached. When you commit you add the `-s` flag to your commit. For example:

```
git commit -s -m "test dco with -s"
```

This will append a statement to your commit comment that attests to the DCO. GitHub has built in the `-s` option to its command line since use of this is so pervasive. The promise is very basic, certifying that you know that you have the right to commit the code. Please read the full statement here.

5.3.3 Questions

Please refer to our *FAQs*.

5.3.4 Developer Guide

Tool Philosophy

OpenROAD is a tool to build a chip from synthesizable RTL (Verilog) to completed physical layout (manufacturable, tapeout-clean GDSII).

The unifying principle behind the design of OpenROAD is for all of the tools to reside in one tool, with one process, and one database. All tools in the flow should use Tcl commands exclusively to control them instead of external "configuration files". File-based communication between tools and forking processes is strongly discouraged. This architecture streamlines the construction of a flexible tool flow and minimizes the overhead of invoking each tool in the flow.

Tool File Organization

Every tool follows the following file structure, grouping sources, tests and headers together.

- `src/` This folder contains the source files for individual tools.

src	Purpose
CMakeLists.txt	add_subdirectory for each tool
tool/src	sources and private headers
tool/src/CMakeLists.txt	tool specific CMake file
tool/include/tool	exported headers
tool/test	tool tests
tool/regression	tool unit tests

- OpenROAD repository: This folder contains the top-level files for overall compilation. OpenROAD uses swig that acts as a wrapper for C/C++ programs to be callable in higher-level languages, such as Python and Tcl.

OpenROAD	Purpose
CMakeLists.txt	top-level CMake file
src/Main.cc	main file
src/OpenROAD.cc	OpenROAD class functions
src/OpenROAD.i	top-level swig, includes, tool swig files
src/OpenROAD.tcl	basic read/write lef/def/db commands
include/ord/OpenROAD.hh	OpenROAD top-level class, has instances of tools

Some tools such as OpenSTA are submodules, which are simply subdirectories in `src/` that are pointers to the git submodule. They are intentionally not segregated into a separate module.

The use of submodules for new code integrated into OpenROAD is strongly discouraged. Submodules make changes to the underlying infrastructure (e.g., OpenSTA) difficult to propagate across the dependent submodule repositories.

Where external/third-party code that a tool depends on should be placed depends on the nature of the dependency.

- Libraries - code packaged as a linkable library. Examples are `tcl`, `boost`, `zlib`, `eigen`, `lemon`, `spdlog`.

These should be installed in the build environment and linked by OpenROAD. Document these dependencies in the top-level `README.md` file. The `Dockerfile` should be updated to illustrate where to find the library and how to install it. Adding libraries to the build environment requires coordination with system administrators, so that continuous integration hosts ensure that environments include the dependency. Advance notification should also be given to the development team so that their private build environments can be updated.

Each tool CMake file builds a library that is linked by the OpenROAD application. The tools should not define a `main()` function. If the tool is Tcl only and has no C++ code, it does not need to have a CMake file. Tool CMake files should **not** include the following:

- `cmake_minimum_required`
- `GCC_COVERAGE_COMPILE_FLAGS`
- `GCC_COVERAGE_LINK_FLAGS`
- `CMAKE_CXX_FLAGS`
- `CMAKE_EXE_LINKER_FLAGS`

None of the tools have commands to read or write LEF, DEF, Verilog or database files. For consistency, these functions are all provided by the OpenROAD framework.

Tools should package all of their state in a single class. An instance of each tool class resides in the top-level OpenROAD object. This allows multiple tools to exist at the same time. If any tool keeps state in global variables (even static), then only one tool can exist at a time. Many of the tools being integrated were not built with this goal in mind and will only work on one design at a time.

Each tool should use a unique namespace for all of its code. The same namespace should be used for Tcl functions, including those defined by a swig interface file. Internal Tcl commands stay inside the namespace, and user visible Tcl commands should be defined in the global namespace. User commands should be simple Tcl commands such as `global_placement` that do not create tool instances that must be based to the commands. Defining Tcl commands for a tool class is fine for internal commands, but not for user visible commands. Commands have an implicit argument of the current OpenROAD class object. Functions to get individual tools from the OpenROAD object can be defined.

Initialization (C++ tools only)

The OpenROAD class has pointers to each tool, with functions to get each tool. Each tool has (at a minimum) a function to make an instance of the tool class, an initialization function that is called after all of the tools have been made, and a function to delete the tool. This small header does **not** include the class definition for the tool so that the OpenROAD framework does not have to know anything about the tool internals or include a gigantic header file.

`MakeTool.hh` defines the following:

```
Tool *makeTool();
void initTool(OpenRoad *openroad);
void deleteTool(Tool *tool);
```

The `OpenRoad::init()` function calls all of the `makeTool` functions and then all of the `initTool()` functions. The `init` functions are called from the bottom of the tool dependencies. Each `init` function grabs the state it needs out of the `OpenRoad` instance.

Commands

Tools should provide Tcl commands to control them. Tcl object based tool interfaces are not user-friendly. Define Tcl procedures that take keyword arguments that reference the `OpenRoad` object to get tool state. OpenSTA has Tcl utilities to parse keyword arguments (`sta::parse_keyword_args`). See `OpenSTA/tcl/*.tcl` for examples. Use swig to define internal functions to C++ functionality.

Tcl files can be included by encoding them in CMake into a string that is evaluated at run time (See `Resizer::init()`).

Errors

Tools should report errors to the user using the `ord::error` function defined in `include/openroad/Error.hh`. `ord::error` throws `ord::Exception`. The variables `ord::exit_on_error` and `ord::file_continue_on_error` control how the error is handled. If `ord::exit_on_error` is `true` then OpenROAD reports the error and exits. If the error is encountered while reading a file with the `source` or `read_sdc` commands and `ord::file_continue_on_error` is `false` then no other commands are read from the file. The default value is `false` for both variables.

Test

Each "tool" has a `/test` directory containing a script named `regression` to run "unit" tests. With no arguments it should run default unit tests.

No database files should be in tests. Read LEF/DEF/Verilog to make a database.

The regression script should not depend on the current working directory. It should be able to be run from any directory. Use filenames relative to the script name rather the current working directory.

Regression scripts should print a concise summary of test failures. The regression script should return an exit code of 0 if there are no errors and 1 if there are errors. The script should **not** print thousands of lines of internal tool information.

Regression scripts should pass the `-no_init` option to `openroad` so that a user's `init` file is not sourced before the tests runs.

Regression scripts should add output files or directories to `.gitignore` so that running does not leave the source repository "dirty".

The Nangate45 open-source library data used by many tests is in `test/Nangate45`. Use the following command to add a link in the tool command:

```
cd src/<tool>/test
ln -s ../../../test/Nangate45
```

After the link is installed, the test script can read the Liberty file with the command shown below.

```
read_liberty Nangate45/Nangate45_typ.lib
```

Building

Instructions for building are available *here*.

Example of Adding a Tool to OpenROAD

The patch file "AddTool.patch" illustrates how to add a tool to OpenROAD. Use the following commands to add a sample tool:

```
patch -p < docs/misc/AddTool.patch
cd src/tool/test
ln -s ../../../test/regression.tcl regression.tcl
```

This adds a directory `OpenRoad/src/tool` that illustrates a tool named "Tool" that uses the file structure described above and defines a command to run the tool with keyword and flag arguments as illustrated below:

```
> toolize foo
Helping 23/6
Gotta positional_argument1 foo
Gotta param1 0.000000
Gotta flag1 false

> toolize -flag1 -key1 2.0 bar
Helping 23/6
Gotta positional_argument2 bar
Gotta param1 2.000000
Gotta flag1 true

> help toolize
toolize [-key1 key1] [-flag1] positional_argument1
```

Documentation

Tool commands should be documented in the top-level OpenROAD `README.md` file. Detailed documentation should be the `tool/README.md` file.

Tool Flow Namespace

Tool namespaces are usually three-lettered lowercase letters.

- Verilog to DB (*dbSTA*)
- OpenDB: Open Database (*odb*)
- TritonPart: constraints-driven paritioner (*par*)
- Floorplan Initialization (*ifp*)
- ICeWall chip-level connections (*pad*)
- I/O Placement (*ppl*)
- PDN Generation (*pdn*)
- Tapcell and Welltie Insertion (*tap*)

- Triton Macro Placer (*mpl*)
- Hierarchical Automatic Macro Placer (*mpl2*)
- RePlAce Global Placer (*gpl*)
- Gate resizing and buffering (*rsz*)
- Detailed placement (*dpl*)
- Clock tree synthesis (*cts*)
- FastRoute Global routing (*grt*)
- Antenna check and diode insertion (*ant*)
- TritonRoute Detailed routing (*drt*)
- Metal fill insertion (*fin*)
- Design for Test (*dst*)
- OpenRCX Parasitic Extraction (*rcx*)
- OpenSTA timing/power report (*sta*)
- Graphical User Interface (*gui*)
- Static IR analyser (*psm*)

Tool Checklist

Tools should make every attempt to minimize external dependencies. Linking libraries other than those currently in use complicates the builds and sacrifices the portability of OpenROAD. OpenROAD should be portable to many different compiler/operating system versions and dependencies make this vastly more complicated.

1. OpenROAD submodules reference tool `openroad` branch head. No git `develop`, `openroad_app`, or `openroad_build` branches.
2. Submodules used by more than one tool belong in `src/`, not duplicated in each tool repo.
3. `CMakeLists.txt` does not use add_compile_options include_directories link_directories link_libraries. Use target_ versions instead. See tips here.
4. `CMakeLists.txt` does not use glob. Use explicit lists of source files and headers instead.
5. `CMakeLists.txt` does not define `CFLAGS` `CMAKE_CXX_FLAGS` `CMAKE_CXX_FLAGS_DEBUG` `CMAKE_CXX_FLAGS_RELEASE`. Let the top level and defaults control these.
6. No `main.cpp` or main procedure.
7. No compiler warnings for GCC or Clang with optimization enabled.
8. Does not call `flute::readLUT` (called once by `openroad`).
9. Tcl command(s) documented in top level `README.md` in flow order.
10. Command line tool documentation in tool README.
11. Conforms to Tcl command naming standards (no camel case).
12. Does not read configuration files. Use command arguments or support commands.
13. `.clang-format` at tool root directory to aid foreign programmers.
14. No `jenkins/`, `Jenkinsfile`, `Dockerfile` in tool directory.

15. `regression` script named `test/regression` with no arguments that runs tests. Not `tests/regression-tcl.sh`, not `test/run_tests.py` etc.
16. `regression` script should run independent of current directory. For example, `../test/regression` should work.
17. `regression` should only print test results or summary, not belch 1000s of lines of output.
18. Test scripts use OpenROAD tcl commands (not `itcl`, not internal accessors).
19. `regression` script should only write files in a directory that is in the tool's `.gitignore` so the hierarchy does not have modified files in it as a result or running the regressions.
20. Regressions report no memory errors with `valgrind` (stretch goal).
21. Regressions report no memory leaks with `valgrind` (difficult).

Code Linting and Formatting

OpenROAD uses both `clang-tidy` and `clang-format` to perform automatic linting and formatting whenever a pull request is submitted. To run these locally, please first setup Clang Tooling using this guide. Thereafter, you may run these commands:

```
cmake . -B build  # generate build files
# typically only run these commands on files you changed.
clang-tidy -p ./build source_file.cpp
clang-format -i -style=file:.clang-format source_file.cpp
```

Guidelines

1. Internally, the code should use `int` for all database units and `int64_t` for all area calculations. Refer to this link for a more detailed writeup on the reasons why this approach is preferred. The only place that the database distance units should appear in any program should be in the user interface, as microns are easier for humans than DBUs.

5.3.5 Coding Practices

List of coding practices.

Note: This is a compilation of many idioms in OpenROAD code that are considered undesirable.

C++

Practice #1

Don't comment out code, instead remove it. `git` provides a complete history of the code if you want to look backwards. Huge chunks of commented-out code make it difficult to read.

Practice #2

Don't use prefixes on function names or variables. That's what namespaces are for.

```
namespace fr {
  class frConstraint
  class frLef58CutClassConstraint
  class frShortConstraint
  class frNonSufficientMetalConstraint
  class frOffGridConstraint
  class frMinEnclosedAreaConstraint
  class frMinStepConstraint
  class frMinimumcutConstraint
  class frAreaConstraint
  class frMinWidthConstraint
  class frLef58SpacingEndOfLineWithinEndToEndConstraint
  class frLef58SpacingEndOfLineWithinParallelEdgeConstraint
  class frLef58SpacingEndOfLineWithinMaxMinLengthConstraint
  class frLef58SpacingEndOfLineWithinConstraint
  class frLef58SpacingEndOfLineConstraint
}
```

Practice #3

Namespaces should be all lower case and short. This is an example of a poor choice: `namespace TritonCTS`

Practice #4

Don't use `extern` on function definitions. It is pointless in a world with prototypes.

```
namespace fr {
  extern frCoord getGCELLGRIDX();
  extern frCoord
  getGCELLGRIDY();
  extern frCoord getGCELLOFFSETX();
  extern frCoord
  getGCELLOFFSETY();
}
```

Practice #5

Don't use prefixes on file names. That's what directories are for.

```
frDRC.h frDRC_init.cpp frDRC_main.cpp frDRC_setup.cpp frDRC_util.cpp
```

Practice #6

Don't name variables `theThingy`, `curThingy` or `myThingy`. It is just distracting extraneous verbiage. Just use `thingy`.

```
float currXSize;
float currYSize;
float currArea;
float currWS;
float currWL;
float currWLnoWts;
```

Practice #7

Do not use global variables. All state should be inside of classes. Global variables make multi-threading next to impossible and preclude having multiple copies of a tool running in the same process. The only global variable in `openroad` should be the singleton that Tcl commands reference.

```
extern std::string DEF_FILE;
extern std::string GUIDE_FILE;
extern std::string OUTGUIDE_FILE;
extern std::string LEF_FILE;
extern std::string OUTTA_FILE;
extern std::string OUT_FILE;
extern std::string DBPROCESSNODE;
extern std::string OUT_MAZE_FILE;
extern std::string DRC_RPT_FILE;
extern int MAX_THREADS ;
extern int VERBOSE ;
extern int BOTTOM_ROUTING_LAYER;
extern bool ALLOW_PIN_AS_FEEDTHROUGH;
extern bool USENONPREFTRACKS;
extern bool USEMINSPACING_OBS;
extern bool RESERVE_VIA_ACCESS;
extern bool ENABLE_BOUNDARY_MAR_FIX;
```

Practice #8

Do not use strings (names) to refer to database or sta objects except in user interface code. DEF, SDC, and Verilog all use different names for netlist instances and nets, so the names will not always match.

Practice #9

Do not use continue. Wrap the body in an if instead.

```
// instead of
for(dbInst* inst : block->getInsts() ) {
  // Skip for standard cells
  if (inst->getBBox()->getDY() <= cellHeight) { continue; }
  // code
```

(continues on next page)

(continued from previous page)

```
}
// use
for(dbInst* inst : block->getInsts() ){
  // Skip for standard cells
  if (inst->getBBox()->getDY() > cellHeight) {
    // code
  }
}
```

Practice #10

Don't put magic numbers in the code. Use a variable with a name that captures the intent. Document the units if they exist.

```
referenceHpwl_= 446000000;
coeffV = 1.36;
coeffV = 1.2;
double nearest_dist = 99999999999;
if (dist < rowHeight * 2) {}
for(int i = 9; i > -1; i--) {}
if(design_util > 0.6 || num_fixed_nodes > 0) div = 1;
avail_region_area += (theRect->xUR - theRect->xLL - (int)theRect->xUR % 200 + (int)t␣
↪heRect->xLL % 200 - 200) * (theRect->yUR - theRect->yLL - (int)theRect->yUR % 2000 +␣
↪(int)theRect->yLL % 2000 - 2000);
```

Practice #11

Don't copy code fragments. Write functions.

```
// 10x
int x_pos = (int)floor(theCell->x_coord / wsite + 0.5);
// 15x
int y_pos = (int)floor(y_coord / rowHeight + 0.5);

// This
nets[newnetID]->netIDorg = netID;
nets[newnetID]->numPins = numPins;
nets[newnetID]->deg = pinInd;
nets[newnetID]->pinX = (short *)malloc(pinInd* sizeof(short));
nets[newnetID]->pinY = (short *)malloc(pinInd* sizeof(short));
nets[newnetID]->pinL = (short *)malloc(pinInd* sizeof(short));
nets[newnetID]->alpha = alpha;

// Should factor out the array lookup.
Net *net = nets[newnetID];
net->netIDorg = netID;
net->numPins = numPins;
net->deg = pinInd;
net->pinX = (short*)malloc(pinInd* sizeof(short));
```

(continues on next page)

(continued from previous page)

```
net->pinY = (short *)malloc(pinInd* sizeof(short));
net->pinL = (short *)malloc(pinInd* sizeof(short));
net->alpha = alpha;

// Same here:
if (grid[j][k].group != UINT_MAX) {
  if (grid[j][k].isValid) {
    if (groups[grid[j][k].group].name == theGroup->name)
      area += wsite * rowHeight;
  }
}
```

Practice #12

Don't use logical operators to test for null pointers.

```
if (!net) {
  // code
}

// should be
if (net != nullptr) {
  // code
}
```

Practice #13

Don't use `malloc`. Use `new`. We are writing C++, not C.

Practice #14

Don't use C style arrays. There is no bounds checks for them so they invite subtle memory errors to unwitting programmers who fail to use `valgrind`. Use `std::vector` or `std::array`.

Practice #15

Break long functions into smaller ones, preferably that fit on one screen.

Practice #16

Don't reinvent functions like `round`, `floor`, `abs`, `min`, `max`. Use the std versions.

```
int size_x = (int)floor(theCell->width / wsite + 0.5);
```

Practice #17

Don't use C's stdlib.h `abs`, `fabs` or `fabsf`. They fail miserably if the wrong arg type is passed to them. Use `std::abs`.

Practice #18

Fold code common to multiple loops into the same loop. Each of these functions loops over every instance like this:

```
legal &= row_check(log);
legal &= site_check(log);
for(int i = 0; i < cells.size(); i++) {
  cell* theCell = &cells[i];
  legal &= power_line_check(log);
  legal &= edge_check(log);
  legal &= placed_check(log);
  legal &= overlap_check(log);
}
// with this loop
for(int i = 0; i < cells.size(); i++) {
  cell* theCell = &cells[i];
}
```

Instead make one pass over the instances doing each check.

Practice #19

Don't use `== true`, or `== false`. Boolean expressions already have a value of true or false.

```
if(found.first == true) {
  // code
}
// is simply
if(found.first) {
  // code
}
// and
if(found.first == false) {
  // code
}
// is simply
if(!found.first) {
 // code
}
```

Practice #20

Don't nest if statements. Use && on the clauses instead.

```
if(grid[j][k].group != UINT_MAX)
  if(grid[j][k].isValid == true)
    if(groups[grid[j][k].group].name == theGroup->name)
```

is simply

```
if(grid[j][k].group != UINT_MAX
   && grid[j][k].isValid
   && groups[grid[j][k].group].name == theGroup->name)
```

Practice #21

Don't call return at the end of a function that does not return a value.

Practice #22

Don't use <> to include anything but system headers. Your project's headers should never be in <>.

1. GCC Include Syntax
2. StackOverflow discussion on "filename" vs <filename>

These are all wrong:

```
#include <odb/db.h>
#include <sta/liberty/Liberty.hh>
#include <odb/db.h>
#include <odb/dbTypes.h>
#include <odb/defin.h>
#include <odb/defout.h>
#include <odb/lefin.h>
```

Practice #23

Don't make "include the kitchen sink" headers and include them in every source file. This is convenient but slows the builds down for everyone. Make each source file include just the headers it actually needs.

```
// Types.hpp
#include <sta/liberty/Liberty.hh>
#include <odb/db.h>
#include <odb/dbTypes.h>
// It should be obvious that every source file is not reading def.
#include <odb/defin.h>
// or writing it.
#include <odb/defout.h>
#include <odb/lefin.h>
#include "db_sta/dbNetwork.hh"
#include "db_sta/dbSta.hh"
```

Note this example also incorrectly uses <>'s around OpenROAD headers.

Header files should only include files to support the header. Include files necessary for code in the code file, not the header.

In the example below NONE of the system files listed are necessary for the header file.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <limits.h>

unsigned num_nets = 1000;
unsigned num_terminals = 64;
unsigned verbose = 0;
float alpha1 = 1;
float alpha2 = 0.45;
float alpha3 = 0;
float alpha4 = 0;
float margin = 1.1;
unsigned seed = 0;
unsigned root_idx = 0;
unsigned dist = 2;
float beta = 1.4;
bool runOneNet = false;
unsigned net_num = 0;
```

Practice #24

Use class declarations if you are only referring to objects by pointer instead of including their complete class definition. This can vastly reduce the code the compiler has to process.

```
class Network;
// instead of
#include "Network.hh"
```

Practice #25

Use pragma once instead of `#define` to protect headers from being read more than once. The #define symbol has to be unique, which is difficult to guarantee.

```
// Instead of:
#ifndef __MACRO_PLACER_HASH_UTIL__
#define __MACRO_PLACER_HASH_UTIL__
#endif
// use
#pragma once
```

Practice #26

Don't put `using namespace` inside a function.

Practice #27

Don't nest namespaces.

Practice #28

Avoid `using namespace`. It increases the likelihood of conflicts and doesn't explicity declare what in the namespace is being used. Use `using namespace::symbol;` instead. And especially do not use `using namespace std`.

The following is especially confused because it is trying to "use" the symbols in code that are already in the MacroPlace namespace.

```
using namespace MacroPlace;

namespace MacroPlace { }
```

Practice #29

Use `nullptr` instead of `NULL`. This is the C++ approved version of the ancient C `#define`.

Practice #30

Use range iteration. C++ iterators are ugly and verbose.

```
// Instead of
odb::dbSet::iterator nIter;
for (nIter = nets.begin(); nIter != nets.end(); ++nIter) {
  odb::dbNet* currNet = *nIter;
  // code
}
// use
for (odb::dbNet* currNet : nets) {
  // code
}
```

Practice #31

Don't use end of line comments unless they are very short.

```
for (int x = firstTile._x; x <= lastTile._x; x++) { // Setting capacities of edges␣
↪completely inside the adjust region according the percentage of reduction
  // code
}
```

Practice #32

Don't `std::pow` for powers of 2 or for decimal constants.

```
// This
double newCapPerSqr = (_options->getCapPerSqr() * std::pow(10.0, -12));
// Should be
double newCapPerSqr = _options->getCapPerSqr() * 1E-12;

// This
unsigned numberOfTopologies = std::pow(2, numberOfNodes);
// Should be
unsigned numberOfTopologies = 1 << numberOfNodes;
```

Git

Practice #33

Don't put /'s in `.gitignore` directory names. `test/`

Practice #34

Don't put file names in `.gitignore` ignored directories. `test/results` `test/results/diffs`

Practice #35

Don't list compile artifacts in `.gitignore`. They all end up in the build directory so each file type does not have to appear in `.gitignore`.

All of the following are to be avoided:

Compiled Object files

`*.slo *.lo *.o *.obj`

Precompiled Headers

`*.gch *.pch`

Compiled Dynamic libraries

```
*.so *.dylib *.dll
```

Fortran module files

```
*.mod *.smod
```

Compiled Static libraries

```
*.lai *.la *.a *.lib
```

CMake

Practice #35

Don't change compile flags in `cmake` files. These are set at the top level and should not be overridden.

```
set(CMAKE_CXX_FLAGS "-O3")
set(CMAKE_CXX_FLAGS_DEBUG "-g -ggdb")
set(CMAKE_CXX_FLAGS_RELEASE "-O3")
```

Practice #36

Don't put /'s in CMake directory names. CMake knows they are directories.

```
target_include_directories( ABKCommon PUBLIC ${ABKCOMMON_HOME} src/ )
```

Practice #37

Don't use `glob`. Explictly list the files in a group.

```
# Instead of
file(GLOB_RECURSE SRC_FILES ${CMAKE_CURRENT_SOURCE_DIR}/src/*.cpp)
# should be
list(REMOVE_ITEM SRC_FILES ${CMAKE_CURRENT_SOURCE_DIR}/src/Main.cpp)
list(REMOVE_ITEM SRC_FILES ${CMAKE_CURRENT_SOURCE_DIR}/src/Parameters.h)
list(REMOVE_ITEM SRC_FILES ${CMAKE_CURRENT_SOURCE_DIR}/src/Parameters.cpp)
```

5.3.6 Using the Logging Infrastructure

OpenROAD uses spdlog as part of logging infrastructure in order to ensure a clear, consistent messaging and complete messaging interface. A wrapper formats the prefix in the recommended messaging style and limit. A message format is as follows:

```
<tool id>-<message id>  <Message body>.
```

For example,

```
[INFO ODB-0127] Reading DEF file: ./results/asap7/aes/base/4_cts.def
```

All output from OpenROAD tools should be directed through the logging API to ensure that redirection, file logging and execution control flow are handled consistently. This also includes messages from any third-party tool. Use the 'ord' message ID for third-party tools.

The logging infrastructure also supports generating a JSON file containing design metrics (e.g., area or slack). This output is directed to a user-specified file. The OpenROAD application has a `-metrics` command line argument to specify the file.

Handling Messages

OpenROAD supports multiple levels of severity for message outputs: critical, error, warning, information and debug. These are supported by automatic calls to the logger which will then prefix the appropriate severity type to the message.

C++20 Requirements

In C++20 the logger messages are checked during compile time which introduces restrictions around rutime format strings. See docs.

OpenROAD uses `spdlog` which uses `fmt_lib` under the hood. Below is an example of what is no longer allowed.

In order to make use of runtime format strings, we have introduced a `FMT_RUNTIME` macro in Logger.h. You should use this macro any time you pass a dynamic string as the format string

```
logger_->info("{} {}", a, b); // OK

void blah(std::string template& a) {
  logger_->info(a, c); // Illegal
  logger_->info(FMT_RUNTIME(a), c); // Ok
}
```

Messaging Guidelines

In addition to the proper use of message types, follow the guidelines below to compose messages for clarity, consistency and other guidelines:

Grammar

Start with a capital letter and end with a period, besides well-known exceptions. Use capital letters for file formats and tool proper names, e.g., LEF, DEF, SPICE, FLUTE.

After the first word's capitalization, do not use capital letters (aside from obvious exceptions, such as RSMT, hCut, etc.).

Do not use exclamations. Severity must be communicated by message severity and clear implied or explicit action.

Avoid long, verbose messages. Use commas to list and separate clauses in messages.

Spellcheck all messages using American English spellings.

Use ellipsis `...` only to indicate a pause, as when some tool is running or being initialized.

Abbreviations and Shortcuts

Use single-word versions when well-accepted / well-understood by users and developers. Examples: `stdcell`, `cutline`, `wirelength`, `flipchip`, `padring`, `bondpad`, `wirebond`, `libcell`, `viarule`.

Do not abbreviate or truncate English words; expand for the sake of clarity.

```
Incorrect:   Num, #;  Tot.
```

```
Correct:     Number;  Total
```

Use acceptable, well-understood abbreviations for brevity. Examples: `db`, `tech`, `lib`, `inst`, `term`, `params`, `etc`.

Avoid contractions of action words:

```
Incorrect:   Can't, Can not;  Don't
```

```
Correct:     Cannot;  Do not
```

Actionability

Messages should communicate a clear, implied or explicit action that is necessary for flow continuation or improved quality of results.

```
Example:
A value for core_area must be specified in the footprint specification, or in the␣
↪environment variable CORE_AREA.
```

Clarity

Messages must be clear and complete, so as to communicate necessary and sufficient information and actions. Elaborate specific variables, options, and/or parameters to avoid any ambiguity.

```
Example:
Unrecognized argument $arg, should be one of -pitch, -bump_pin_name, -spacing_to_edge, -
↪cell_name, -bumps_per_tile, -rdl_layer, -rdl_width, -rdl_spacing.
```

Specify objects clearly in the local context:

```
Example:
cutWithin is smaller than cutSpacing for ADJACENTCUTS on layer {}. Please check your␣
↪rule definition.

Incomplete:
Warning: {} does not have viaDef aligned with layer.
```

Make any assumptions or use of default values explicit:

```
Example:
No net slacks found.
Timing-driven mode disabled.

Incomplete, missing default:
Utilization exceeds 100%.
```

Use simple language, and avoid repetitions:

```
Example:
Missing orientation for cell $cell_ref.

Incorrect:
No orientation available for orientation of $cell_ref.
```

Message Types

OpenROAD supports the following levels of severity through the logger: report, debug, information, warning, error and critical.

Report

Report messages are output by the tool in the form of a report to the user. Examples include timing paths or power analysis results.

Example report message:

```
Path startpoint: $startpoint
```

Debug

Debug messages are only of use to tool developers and not to end users. These messages are not shown unless explicitly enabled.

Information

Information messages may be used to report metrics, quality of results, or program status to the user. Any message which indicates runtime problems, such as potential faulty input or other internal program issues, should be issued at a higher status level.

Example information messages:

```
Number of input ports: 47

Running optimization iteration 2

Current cell site utilization: 57.1567%
```

Warning

Warnings should be used to indicate atypical runtime conditions that may affect quality, but not correctness, of the output. Any conditions that affect correctness should be issued at a higher status level.

Example warning messages:

```
Core area utilization is greater than 90%. The generated cell placement may not be␣
↪routable.

14 outputs are not constrained for max capacitance.

Pin 'A[0]' on instance 'mem01' does not contain antenna information and will not be␣
↪checked for antenna violations.
```

Error

Error messages should be used to indicate correctness problems. Problems with command arguments are a good example of where error messages are appropriate. Errors exit the current command by throwing an exception that is converted to an error in Tcl. Errors that occur while reading a command file stop execution of the script commands.

Example error messages:

```
Invalid selection: net 'test0' does not exist in the design.

Cell placement cannot be run before floorplanning.

Argument 'max_routing_layer' expects an integer value from 1 to 10.
```

Critical

Critical messages should be used to indicate correctness problems that the program is not able to work around or ignore, and that require immediate exiting of the program (abort).

Example critical messages:

```
Database 'chip' has been corrupted and is not recoverable.

Unable to allocate heap memory for array 'vertexIndices'. The required memory size may␣
↪exceed host machine limits.

Assertion failed: 'nodeVisited == false' on line 122 of example.cpp. Please file a␣
↪Github issue and attach a testcase.
```

Coding

Each status message requires:

- The three letter tool ID
- The message ID
- The message string
- Optionally, additional arguments to fill in placeholders in the message string

Reporting is simply printing and does not require a tool or message ID. The tool ID comes from a fixed enumeration of all the tools in the system. This enumeration is in `Logger.h`. New abbreviations should be added after discussion with the OpenROAD system architects. The abbreviation matches the C++ namespace for the tool.

Message IDs are integers. They are expected to be unique for each tool. This has the benefit that a message can be mapped to the source code unambiguously even if the text is not unique. Maintaining this invariant is the tool owner's responsibility. To ensure that the IDs are unique, each tool should maintain a file named 'messages.txt' in the top-level tool directory, listing the message IDs along with the format string. When code that uses a message ID is removed, the ID should be retired by removing it from 'messages.txt'. See the utility `etc/find_messages.py` to scan a tool directory and write a `messages.txt` file.

Spdlog comes with the `fmt` library which supports message formatting in a python or C++20 like style.

The message string should not include the tool ID or message ID which will automatically be prepended. A trailing newline will automatically be added, and hence messages should not end with one. Messages should be written as complete sentences and end in a period. Multi-line messages may contain embedded new lines.

Some examples:

```
logger->report("Path startpoint: {}", startpoint);

logger->error(ODB, 25, "Unable to open LEF file {}.", file_name);

logger->info(DRT, 42, "Routed {} nets in {:3.2f}s.", net_count, elapsed_time);
```

Tcl functions for reporting messages are defined in the OpenROAD swig file `OpenRoad.i`. The message is simply a Tcl string (no C++20 formatting).

```
utl::report "Path startpoint: $startpoint"
```

(continues on next page)

(continued from previous page)

```
utl::error ODB 25 "Unable to open LEF file $file_name."

utl::info DRT 42 "Routed $net_count nets in [format %3.2f $elapsed_time]."
```

`utl::report` should be used instead of 'puts' so that all output is logged.

Calls to the Tcl functions `utl::warn` and `utl::error` with a single message argument report with tool `ID UKN` and message `ID 0000`.

Tools use `#include utl/Logger.h` that defines the logger API. The Logger instance is owned by the OpenROAD instance. Each tool should retrieve the logger instance in the tool init function called after the tool make function by the OpenROAD application.

Every tool swig file must include src/Exception.i so that errors thrown by `utl::error` are caught at the Tcl command level. Use the following swig command before `%inline`.

```
%include "../../Exception.i"
```

The logger functions are shown below.

```
Logger::report(const std::string& message,
               const Args&... args)
Logger::info(ToolId tool,
             int id,
             const std::string& message,
             const Args&... args)
Logger::warn(ToolId tool,
             int id,
             const std::string& message,
             const Args&... args)
Logger::error(ToolId tool,
              int id,
              const std::string& message,
              const Args&... args)
Logger::critical(ToolId tool,
                 int id,
                 const std::string& message,
                 const Args&... args)
```

The corresponding Tcl functions are shown below.

```
utl::report message
utl::info tool id message
utl::warn tool id message
utl::error tool id message
utl::critical tool id message
```

Although there is a `utl::critical` function, it is really difficult to imagine any circumstances that would justify aborting execution of the application in a tcl function.

Debug Messages

Debug messages have a different programming model. As they are most often *not* issued the concern is to avoid slowing down normal execution. For this reason such messages are issued by using the debugPrint macro. This macro will avoid evaluating its arguments if they are not going to be printed. The API is:

```
debugPrint(logger, tool, group, level, message, ...);
```

The `debug()` method of the Logger class should not be called directly. No message id is used as these messages are not intended for end users. The level is printed as the message id in the output.

The argument types are as for the info/warn/error/critical messages. The one additional argument is group which is a `const char*`. Its purpose is to allow the enabling of subsets of messages within one tool.

Debug messages are enabled with the tcl command: `set_debug_level <tool> <group> <level>`

Metrics

The metrics logging uses a more restricted API since JSON only supports specific types. There are a set of overloaded methods of the form:

```
metric(ToolId tool,
       const std::string_view metric,
       <type> value)
```

where <type> can be `int, double, string, or bool`. This will result in the generated JSON:

```
"<tool>-<metric>" : value
```

String values will be enclosed in double-quotes automatically.

Converting to Logger

The error functions in `include/openroad/Error.hh` should no longer be included or used. Use the corresponding logger functions.

All uses of the tcl functions ord::error and ord::warn should be updated call the `utl::error/warn` with a tool ID and message ID. For compatibility these are defaulted to `UKN` and `0000` until they are updated.

Regression tests should not have any `UKN-0000` messages in their ok files. A simple grep should indicate that you still have pending calls to pre-logger error/warn functions.

The `cmake` file for the tool must also be updated to include spdlog in the link libraries so it can find the header files if they are not in the normal system directories.

Tip: At UCSD, dfm.ucsd.edu is an example of this problem; it has an ancient version of spdlog in '/usr/include/spdlog'. Use `module` to install spdlog 1.8.1 on dfm.ucsd.edu and check your build there.

```
target_link_libraries(<library_target>
  PUBLIC
    utl
)
```

Useful Information

As tool developers, we can also choose to include useful information to the end user - be it in the form on debugging tips, or solutions to fix the errors/warnings. We compile a list of such errors in this *table*. The good thing about this page is the ability to encode rich formatting using Markdown, enabling you to convey more information than what can be said from the limited messages in code.

To format the information, refer to this sample GRT information file. In addition, make sure you create the corresponding `docs/messages` folder under the `tool` folder, before creating your Markdown file with the corresponding `NUM`.

```
cd src/<tool> && mkdir -p doc/messages
cd doc/messages && touch <NUM>.md
```

OpenROAD Tool List

A full list of tool namespaces can be found *here*.

5.3.7 CI Guide

This document describes the pipelines available to the developers and code maintainers in the Jenkins server. Note that pipelines with the suffix `*-Private` are only available to code maintainers and The OpenROAD Project members as they can contain confidential information. Thus, to access Private pipelines one needs to have authorization to access confidential data and be logged in the Jenkins website.

Below there is a list of the available features. Instructions on how to navigate Jenkins to access these features are available here.

- Find your build through Jenkins website or from GitHub.
- See test status: Pass/Fail.
- Log files for each test.
- Build artifacts to reproduce failures.
- HTML reports about code coverage and metrics.

OpenROAD App

- OpenROAD-Coverage-Public
  - Description: run dynamic code coverage tool `lconv`.
  - Target: master branch.
  - Report link here.
- OpenROAD-Coverity-Public
  - Description: compile and submit builds to Coverity static code analysis tool.
  - Target: master branch.
  - Report link here.
- OpenROAD-Nightly-Public

  - Description: `openroad` unit tests, docker builds, ISPD 2018 and 2019 benchmarks for DRT and large unit tests of GPL.
  - Target: master branch.
- OpenROAD-Public
  - Description: `openroad` unit tests and docker builds.
  - Target: all branches and open PRs.
- OpenROAD-Special-Private
  - Description: for developer testing, runs ISPD 2018 and 2019 benchmarks for DRT and large unit tests of GPL.
  - Target branches: `TR_*`, `secure-TR_*`, `TR-*`, `secure-TR-*`.
- OpenROAD-Private
  - Description: `openroad` unit tests and docker builds.
  - Target: all branches. Note that PRs will be run on public side after "Ready to Sync Public" workflow.

OpenROAD Flow

- Information about OpenROAD Flow CI jobs can be found here

OpenLane

- OpenLane-MPW-CI-Public
  - Description: test projects to older MPW shuttles with newer OpenLane versions.
  - Repo link.
- OpenLane-Public
  - Description: test OpenLane with latest commit from OpenROAD.
  - Repo link.

5.4 Contributor Covenant Code of Conduct

5.4.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

5.4.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

5.4.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

5.4.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

5.4.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at complaints@openroad.tools. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

5.4.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

5.4.7 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by Mozilla's code of conduct enforcement ladder.

For answers to common questions about this code of conduct, see the FAQ at https://www.contributor-covenant.org/faq. Translations are available at https://www.contributor-covenant.org/translations.

5.5 FAQs

If you cannot find your question/answer here, please file a GitHub issue to the appropriate repository or start a discussion.

- Issues and bugs:
  - OpenROAD: https://github.com/The-OpenROAD-Project/OpenROAD/issues
- Discussions:
  - OpenROAD: https://github.com/The-OpenROAD-Project/OpenROAD/discussions

5.5.1 How can I contribute?

Thank you for your willingness to contribute. Please see the *Getting Involved* guide.