OpenROAD Limitations and Restrictions

Global

- Only tested/validated on CentOS 6/7
- Development and support in 1H2019 has focused solely on the TSMC65LP process and 12-track standard cells + SRAMs from ARM,
- Exploration of support for FreePDK45 and TSMC16FFC has been performed. Partial support for Nangate45 cells is available
- Restoration of router support for, e.g., open academic ISPD18/19 LEFs and testcases are planned by August 2019.
- Use compiled binaries from the release folder
- Several tools require **a single merged lef** (see details of tool steps, and merging script, below) **with the following modifications**.
    - **Exclusion of cells with difficult pin access.** There are several cells that have closely packed pins that may need to be excluded from the entire flow to achieve good drc results. In general, it is safest to strip sub-X1, AOI22X1 and other known difficult-to-access masters from the library.
    - **Cell padding.** Because of a current lack of integrated global routing under the hood of placement (NCTUgr binary no longer invoked by default), cell padding is strongly recommended. E.g., padding by 5cpp to the left. A script to perform this task is here

Synthesis (Yosys+ABC)

- Requires a single merged lef. Find a utility script to perform merging here.
- Does not support full sdc suite of command. Only supports create_clock, set_max_fanout and set_max_transition
- Yosys often produces netlists with “don’t care” assignments that other tools don’t like. Run the following command to remove those assignments

```
sed -E "/assign(.*)1'(.*)x;/d"  input.v > output.v
```

- Yosys does not support a user provided “dont_use” cell list. There are several cells that have pins that are difficult to access, particularly when particular instance abutments occur in the placement. See “Exclusion of cells with difficult pin access” and “Cell padding”, above.
- If using physical synthesis, see requirements for Floorplanning and Placement that may be called from Yosys.

Floorplanning (Resizer, ioPlacer, MacroPlacer, PDN, tapcell)

- Recommended placement density utilization is 40 percent. Can be easily be achieved with the “utilization” argument of verilog2def.
- verilog2def requires a tracks.info file. See more information here
- **Cell padding** is required for good routability. See above.
- TritonFP only supports vertical power stripes on macros.
- Use prescribed power layout in the PDN generation scripts (PDN.cfg files). The users must specify all the width, spacing and pitch numbers for each layer in PDN.cfg. Also, via types, rules and arrays should be defined in PDN.cfg. The PDN.cfg is customized and hard-coded for any given technology, library and BEOL stack (e.g., TSMC65LP technology, Arm 12T cells, Arm SRAMs, 9-layer metal …).
- Similarly, users must provide IP_local.cfg and IP_global.cfg config files which are hard-coded for particular IPs that are to be instantiated in the design.

Placement (RePlAce + OpenDP, with Resizer for buffering/sizing)

- RePlAce requires flute (PORT9.dat, POST9.dat and POWV9.dat) data files in the current working directory (./).
- Resizer requires flute (PORT9.dat, POST9.dat and POWV9.dat) data files in (../etc).
- Resizer requires resistance and capacitance information.
- Resizer also needs to adhere to the same dont_use list

Clock Tree Synthesis (TritonCTS, with OpenDP legalization)

- Requires one time library characterization for process
- Add a "Dummy buffer" macro to your technology .lef file. You may do this by duplicating any buffer macro and renaming it as "DUMMY".
- Clock sinks should be positioned in an area close to a standard row where a buffer can be placed

Route (UTD-BoxRouter + TritonRoute)

- **It is helpful to restrict the global route guide generation to lower layers**, as possible. Detailed routing will have difficulty with DRC-correct routes that use wide metals (e.g., if boundary pins are defined on such layers), because of the complexity of DRC checks. Current re-costing of ripup-reroute and maze search does not handle this well. In a 65nm 9-layer stack, routing is most successful when the route guide generation is restricted to the first 7 layers.
- **Gcell dimensions should be neither too large** (poor guidance to the detailed router, more complex track assignment and other steps in detailed route) **nor too small** (many vertices in the global routing gridgraph will slow the mathematical programming solver in

the global router). In various technologies, around two cell row heights is a good gcell dimension. At least one industry tool auto-sets the gcell dimension to 15 M3 pitches, which is reasonable for our detailed router. In a 12T, 65nm enablement, we have used gcell size 6000x6000.

- Replace all instances of "PROPERTY LEF58_SPACING ENDOFLINE“ with just ”SPACING ENDOFLINE”
- From 65LP LEF: antenna rules are not checked / fixed.
- Restoration of detailed router support for, e.g., open academic ISPD18/19 LEFs and testcases is planned by August 2019.

Layout Finishing

- Currently restricted to the DEF-to-GDS (and, GDS merge) step. For example, OpenROAD tools do not yet perform dummy fill insertion or metal fill.
- The DEF-to-GDS step requires a Magic tech file. See more information here