

Instructions for LEF File Generation Process

The abstract generator program *abstract* is used for generating LEF files. However, it is used only for generating the part of LEF files which partially describes the geometry of the cells, and not the complete LEF file. The technology description part of the LEF file must be created manually.

A LEF file describing a library has two parts:

1. The technology description (“header”) part, describing:
 - i. The layers available in the technology. Only layers involved in the PNR process need to be included.
 - ii. Part of design rules which affects PNR operation, such as minimum metal width and separation. It does not usually have to include, for example, rules for separation between n-wells, as wells are not used as wires and usually cells are designed such that no well separation violation should occur even between adjacent cells.
 - iii. Library designer-defined routing rules, such as the chosen value of routing pitch, the preferred directions for metal tracks, or the geometric description of the via used (here the term “via” is used to also include the metal extensions needed in both layers of metals connected to the cut layer).
 - iv. (optional) Electrical properties of the layers in the library, such as maximum current per cut, unit square resistance for the metal layer used.
2. The cell description part, describing the geometries comprising each cell:
 - i. The shape and size of cells, as defined by their respective boundaries
 - ii. The location of pins, and the layer those pins rest on, as well as geometric description of other shapes in the same nodes
 - iii. Detailed descriptions of obstructions, namely shapes in conducting layers which do not belong to any particular pins, but prohibit the passage of routing tracks in the same layer.

The first part (technology description) must be generated either manually or with some other tool. It is required by *abstract* as an input for generation of the second part. The generation of geometric description of even a cell of modest complexity, such as a flip-flop with set/reset, is tedious and error-prone if performed manually, hence the use of *abstract* or other similar tools is mandatory.

Steps involved in generating LEF file:

1. Exporting the GDS files from Virtuoso – for use by *abstract* LEF generator.
2. Creation of technology description (“header”) part of LEF file
3. Importing layout (in GDS format) into *abstract*.
4. Generating the complete LEF file (with cell descriptions).

In detail, this is performed with the following steps, which could also be found in

<cadence-installation-dir>/<SOC-installation-dir>/doc/abstract/abstractTOC.obk,
which could be viewed using Cadence help browser *openbook*.

Have the Technology LEF file ready.

Exporting the GDS2 file

1. Export the layouts in GDS2 format from icfb. Note that for pad cells, we must use the dummy pad cells (the one with only pin locations and names, see library vtvtdummymosispad25) – using the real pins may cause *abstract* to crash.

Header LEF file generation

1. Using a valid technology file for the LEF generation (or the vtv_tsmc250.tf file in the release package), create a new library using Cadence CIW->Technology File Manager->New...
 - The new lib is vtv_tsmc250_techfile
 - Attach the vtv_tsmc250 and pad library to vtv_tsmc250_techfile
 - CIW->Technology File Manager->Attach
 - Design Library: vtv_tsmc250 or pad library
 - Technology Library: vtv_tsmc250_techfile
2. Extract LEF base information from new tech file: From the modified technology file, extract the LEF base information by doing in CIW File->Export->LEF... The file vtv_tsmc250_base.lef is generated. It contains the “header” part of the LEF file.
3. Import the LEF file right back into library vtv_tsmc250_techfile. You can CIW->Technology File Manager->Dump the techfile to make sure that all your changes have been made in the techfile. The CIW window should say: Done reading file. Now you can use the vtv_tsmc250 library in abstract.

Complete LEF file generation

4. Open *abstract* again, with the following command:
abstract &
All operations should be done on Abstract window, and not on PCW window.
5. Choose **File** → **Library**. This should open the abstract library, or create one if it does not already exist.
6. Import the layout file (in GDSII format) : **File** → **Import** → **Layout**. Import the GDSII layout file. Choose “No Mapping” – this will preserve the case in the names of the instances inside the cells. Note that this works best if the cells are flattened. Note: that corner cells must go to “Corner” bin; other pads should go to the “Pad” bin; core cells go to the “Core” bin.
7. Select the cells for which LEF files need to be generated. To select all of them use **Select** → **All**.

Defining pins:

8. Select **Flow** → **Pins**. Then in the new window specify the layers in which labels for pins are drawn. For example, if your pins are all metal1 pins and they are given label in metal2, specify (metal2 metal1) in the topmost text box. The statement means that for all labels in metal2, use the name specified by the label for the metal1 shape over which the label's origin is placed. For our library we use the following three lines: (text (metal2 drawing) (metal1 drawing)) for core; for others, (text metal1) (res_id metal2) (cap_id (metal5 drawing) (metal4 drawing) (metal3 drawing)). Note that power pin names are vdd, vdd1, vdd2, vddm1, and vddm2; ground pin names are vss, vss1, vss2, vssm1, and vssm2.
9. Select **Boundary** tab – and check whether any boundary location adjustments are needed. For core cells, no adjustments will be needed; for pad and corner cells, the size must be adjusted to 100×300 μm and 300×300 μm, respectively.
10. Press **Run** button and wait until pin generation step is completed. Most likely, here will be warnings for parts of the cells being outside the boundary (due to parts of n or p-well which lies outside the border). They could be just ignored, as they will be abutted with the wells of the same polarity inside the adjacent cells during the placement step.

Pin and obstruction extraction:

11. Choose **Flow** → **Extract** → **Run**.

Abstract generation:

12. Choose **Flow** → **Abstract**.
13. Choose the **Overlap** tab.
14. Choose **as needed**.
15. Choose the **Grid** tab – and correct any error in offset or routing grid if necessary.
16. Seems like abstract tends to get it wrong here.
17. Press **Run**.
18. After all is done, export the resulting LEF file.
19. Append the cells' LEF information into the header LEF file.

Postprocessing:

For all pins that are used for power or ground, the SHAPE must be defined. The only exception is the supply / ground for core cells provided by power or ground pads.

For core cells:

The SHAPE property should be defined as ABUTMENT. Hence, lines such as:

```
USE POWER ;
```

or

```
USE GROUND ;
```

should become

```
USE POWER ; SHAPE ABUTMENT ;
```

or

USE GROUND ; SHAPE ABUTMENT ;

respectively. This is usually already performed automatically by abstract generator. If not, it has to be performed manually.

For pad or corner cells:

This is almost never performed manually by the abstract generator. Here the correct property shape for VDD/ground ring pin is FEEDTHRU. Hence, the aforementioned two lines become:

USE POWER ; SHAPE FEEDTHRU ;

USE GROUND ; SHAPE FEEDTHRU ;

respectively.

Miscellaneous tips:

– Labels for pins should be in a nonconducting, physically nonexistent layers (such as text or metal labels) so that would not cause any design rule violations if design rule check (DRC)

needs to be performed inside a layout editor session. While such a DRC error is actually meaningless and can simply be ignored, it may lead to confusion.

– Since pad cells are usually very large, it is likely to be impractical to use a full pad cell for LEF file generation – using the full pad may even cause *abstract* to crash. It is probably better to simply use a dummy cell representing only the outermost edges of pin locations, using thin wires – the same width as specified for normal tracks.