Application Note

Using DRC Error Database to Analyze LVL Run Results

Introduction

Expert layout editor offers the following functions: LVL (Layout Versus Layout), DRC (Design Rule Check) and Lisa (Language for Interfacing Applications, special-purpose language for Expert). Only a license for Expert is required.

This note describes how to use the DRC error navigation tool to analyze results of LVL runs.

Overview

LVL compares and detects the differences between two cells in one project. These differences can be saved in the error database, as errors on a specified cell.

The current version cannot support the command line for LVL, so a manual operation is required using the GUI.

After LVL execution, automatic operation using Lisa is available.

LVL can integrate error layers into one layer or keep them as-is. In the case of the latter, it cannot output from the error database with the Expert license alone, so it is necessary to choose the integration option.

LVL execution

LVL can specify cells which exist in the current library or activated reference library. In the case of using a GDSII file, it is necessary to set the reference library which was converted an to ELD file, or import to the current directory.

This can be called from the Expert menu-bar.

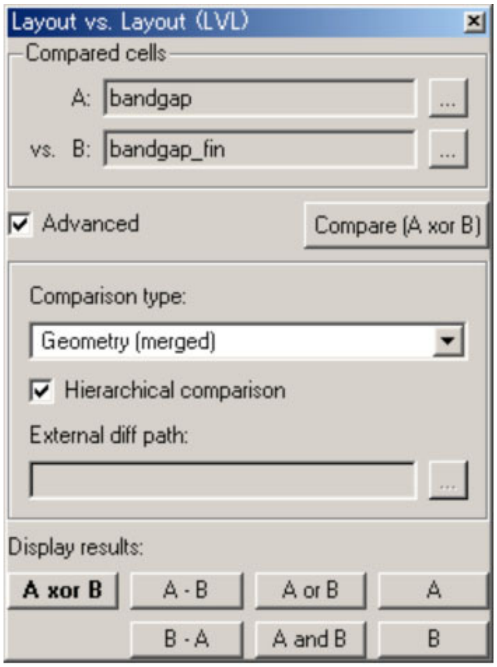
[Verification]

>> [Layout vs. Layout]

Geometry (merged) means merge individual shapes.

Hierarchical Comparison means using geometry from all levels of hierarchy.

There are some logical operations in LVL, and when checking for differences between two cells, the XOR operation should be used.



Layout vs. Layout (LVL)

Compared cells

A: bandgap

vs. B: bandgap_fin

☑ Advanced

Comparison type: Geometry (merged)

☑ Hierarchical comparison

External diff path:

Display results:

Figure 1. LVL setup screen.

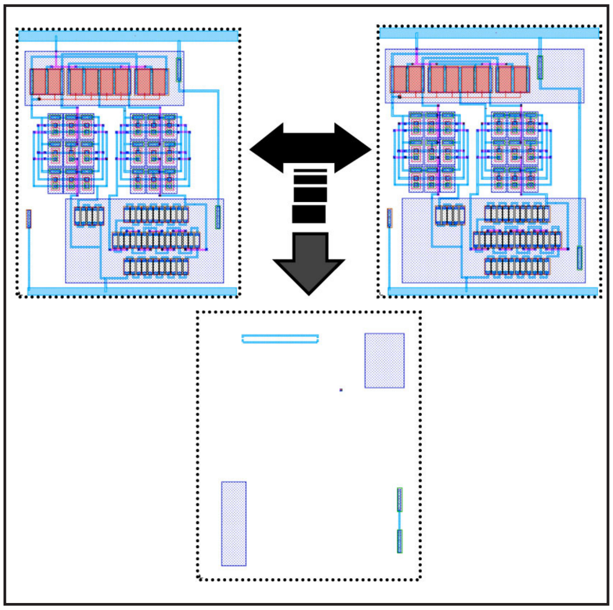


Figure 2. An example of an error database created as a result of an LVL error.

DRC Error Database from Detected Cell

The detected LVL difference is stored as temporary cell data, so it is necessary to save it as a new cell name.

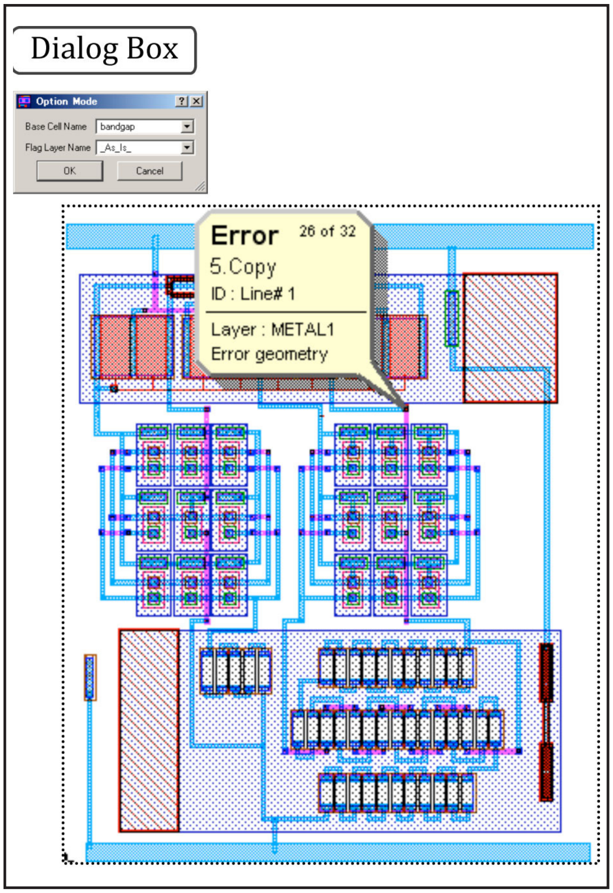
A prepared Lisa script can be used to move temporary geometry data into DRC error database. At that time, all available DRC error inspection functions can be used to review differences between layouts.

The detected differences can be checked like a DRC error. Also, a different layer name is described in a pop-up flag if the flag layer is set to "_As_Is_".

Syntax : lvl_after

XI > lvl_after [/base = A|B /flag=value]

Run lvl_after from the XI command line. A dialog box will appear where the user can specify one or two cells as a base cell and pick a layer name from the drop-down list. The command is canceled if the layer does not exist.



Base Cell Name: bandgap

Flag Layer Name: _As_Is_

Error 26 of 32

5.Copy

ID : Line# 1

Layer : METAL1

Error geometry

Figure 3. The sample of description errors.

Conclusion

All functions introduced in this note are available with an Expert license alone. It is required to integrate into one layer if there are more than 10 different kinds of layers.

LVL execution is performed in GUI mode, so the performance of LVL will be worse if the design has larger scaled integration.

The performance of the output error database depends on the number of detected differences.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!  Output Error DataBase after LVL             !
!      *Presented by SILVACO Japan Co., Ltd.   !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
define command/replace “lvl_after”;
define action
parameter base
parameter flag
do begin

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! make “drop-down-list” of two cells. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  cur_cell = get_edited_cell_name();
  cell_buf = buffer_create();
  buffer_insert_at_end(cell_buf, &cur_cell[10..-1]);
  loop begin
    if(cell_buf.at_end) then (leave loop);
    buf_move = buffer_match(cell_buf, “_xor_”);
    if(buf_move EQL False) then (buffer_advance(cell_buf, 1));
    if(buf_move EQL True) then begin
      buffer_advance(cell_buf, -5);
      buffer_replace(cell_buf, cell_buf.index, (cell_buf.index+4), “¥t”);
      leave loop;
    end;
  end;
  seq_cell = {(&cell_buf)[1..cell_buf.index-1], (&cell_buf)[(cell_buf.index+1)..-1]};

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! make “drop-down-list” of all layers. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!   (Except for the Derive Layer, Datatype=/0 & GDSnumber<0.  !!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  layers = get_layer_list();
  c=1;
  loop begin
    if(c GTR layers.size) then (leave loop);
    if((get_gds_datatype(&layers[c]) NEQ 0) OR (get_gds_number(&layers[c]) LSS 1))
      then(seq_remove_ith(layers, c)) else(c=c+1);
  end;
  layers_buf = buffer_create();
  buffer_insert_at_end(layers_buf, &layers);
  loop begin
    if(layers_buf.at_end) then (leave loop);
    buffer_scan(layers_buf, “¥n”);
    buffer_replace(layers_buf, layers_buf.index, layers_buf.index, “¥t”);
    buffer_advance(layers_buf, 1);
  end;

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Action. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  if(&base EQL “A”) then (cell_name = seq_cell[1])
  elseif(&base EQL “B”) then (cell_name = seq_cell[2])
  else (cell_name = “”);
  layer_name = “_As_Is_”;
  buffer_insert(layers_buf, 1, (&layer_name &”¥t”));

  if((&base EQL “”) OR (flag EQL 0)) then begin
    t1={ &cell_buf , “Base Cell Name”};
    t2={ &layers_buf , “Flag Layer Name”};
    tt={t1,t2};
    tt1={“ Option Mode “};
    vars1 = (form create (tt) (tt1));

    if(&vars1.type NEQ “Null”) then begin      !!! Form
      cell_name  = vars1[1];
      layer_name = vars1[2];
    end; !!!!!!! Form Cancel !!!!!!!
  end;

    cell status /rw;
    select all;
    tools cutbyvertex /all /selected /regionvertices=512 /wirevertices=256;
    copy selection;
!!!! Version 4.4.X !!!!
    all_shapes = (find objects (SEARCH_ANY_SHAPE) /selected /seq_output);
    x_pos = (all_shapes[1].bbox).xpos;
    y_pos = (all_shapes[1].bbox).ypos;
    if(all_shapes.size GEQ 2) then begin
      l = 1;
      loop begin
        l = l + 1;
        if(l GTR all_shapes.size) then (leave loop);
        if((all_shapes[l].bbox).xpos LSS x_pos) then (x_pos = (all_shapes[l].bbox).xpos);
        if((all_shapes[l].bbox).ypos LSS y_pos) then (y_pos = (all_shapes[l].bbox).ypos);
      end;
    end;
!!!!!!!!!!!!!!!!!!!!!!!
    cell close;

  if((cell_exists(cell_name) EQL True) AND ((layer_exists(layer_name) EQL True) OR (&layer_name EQL “_As_Is_”))) then begin     !!! Execution
    if(cell_exists(“lvl_result”) EQL True) then (cell delete “lvl_result”);
    cell new “lvl_result”;
    cell open “lvl_result”;
    instance 0 0 /cell = (cell_name);
    if(&layer_name EQL “_As_Is_”) then begin
      paste selection (x_pos) (y_pos);         !!! Version 4.4.X
!      paste selection;                        !!! Version 4.5.X
      all_layers = get_layer_list();
      m = 0;
      n = 0;
      check = “”;
      loop begin
        m = m + 1;
        if(m GTR all_layers.size) then (leave loop);
        layer_count = get_layer_object_count( (get_edited_cell_name()),  (all_
layers[m]));

        if(layer_count GTR 0 ) then begin
          n = n + 1;
          if(n EQL 10) then begin
            ret = message_box(“There are more than 10 different kinds of layers.¥n”
                         &”Guardian-DRC license is required.¥n¥n”
                         &”Proceed to execute ?”, {});
            if(ret EQL false) then (leave loop);
          end;
          check = &check &”Copy: Layer=” &all_layers[m] &”;”;
        end;
        if(m EQL all_layers.size) then (drc command (check) /cell); ! /selection
      end;
    end
    else begin
      paste layer (layer_name);
      merge selection;
      tools cutbyvertex /all /selected /regionvertices=512 /wirevertices=256;
      zoom /all;
      check = “Copy: Layer=” &layer_name &”;”;
      drc command (check) /cell; ! /selection
    end;

  end;
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

end; !!!!!!! define end  !!!!!!!
define argument base /named /coerce_to=(String) /default=”” /is_default;
define argument flag /named /coerce_to=(Integer) /default=0 /is_default;
complete command;
```