Cadence Design Systems, Inc.

Rapid Adoption Kit (RAK)

Power and Rail Analysis

Using

Voltus IC Integrity

Click to get the test case

Voltus RAK Lab data

Product Version – Voltus 15.1

October, 2015

Contents

Overview

The goal of this Rapid Adoption Kit is to provide a framework of commands that will help you quickly become comfortable with the Voltus IC Power Integrity Solution cockpit. This will cover many aspects of the power and rail analysis:

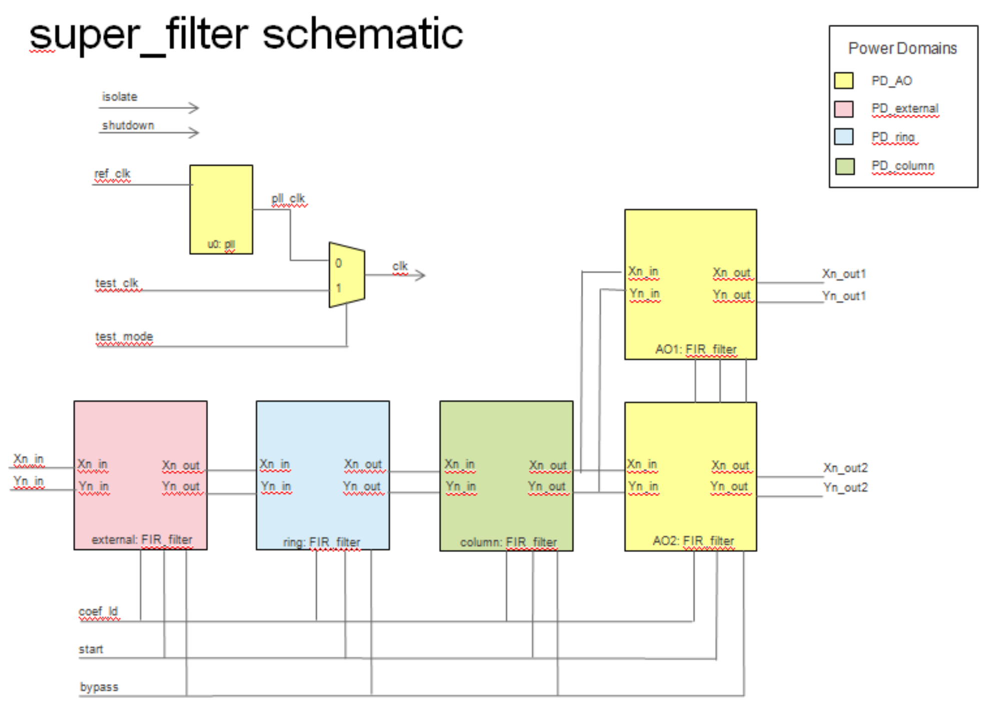
- Library characterization
- Applying static power and piecewise linear current waveforms to control the power of particular instance
- Setting toggle rates to nets
- Vector profiler for analyzing VCD files for windows with high activity, high average power, or high peak power
- Static and dynamic power analysis
- Analyzing current/power plots
- Performing static and dynamic rail analysis
- What-if analysis
- Analyzing various rail analysis plots
- Native power up analysis
- load OA database
- ERA – Early rail analysis
- Effective resistance analysis

Note: The last three features are new to this release.

Please refer to the Voltus User Guide at http://support.cadence.com for more detailed information.

Design Information

The test design is based upon five instantiations of a FIR filter.



The first three filters are connected sequentially and are switchable power domains. The last two stages are parallel and receive the output of the third stage. These two domains are always-on. In total, there are 8653 instances in the design, excluding fill cells. It has been designed with the Cadence Generic 90nm PDK. The only macro in the design is a PLL. The netlist associated with it is bogus so it can be utilized and delivered with the kit.

Voltus Flow Overview

There are three major stages of the power grid analysis in Voltus:

1. Voltus library generation
2. Power analysis to calculate static and dynamic power
3. Static and dynamic rail analysis to report IR / EM results

Library Characterization

In addition to liberty models (timing) and LEF/OA (physical) information used to place/route a design, a prerequisite for rail analysis is a power grid view (PGV) library. In Voltus, there are two types of libraries:

- Technology library
- Power-grid library

Technology library is the minimum requirement for running the rail analysis, and it contains the extraction tech file, filler/decap/powergate identifications, and a tech view for every cell. Power-grid library contains three types of power-grid views (PGV) for each cell: Early/IR/EM. The PGV provides more accuracy to rail analysis, and is recommended to generate power-grid library for every standard cells and macros in the design.

Technology library

- Technology related data
- Tech View
    - Power net consists of power pins only, with no detail inside the cell.
    - Current taps (sinks) are added to the power-pins, and total current is distributed evenly across all the current taps.
    - Area-based cell capacitance is distributed on power-pins evenly.
    - If you define power gate cells in technology library generation, the TECH view can be used for power up power analysis to generate sequence file. However, it cannot be used for power up rail analysis. You need to generate power-grid library for power gate cells.

Power-grid library

- Early power-grid view (Standard cells/Macros)
    - Power net consists of power pins only, with no detail inside the cell.
    - Current tap/sink is attached to each power port, and is evenly distributed among all power and ground pins.
    - Decap value is characterized from spice simulation, and is attached on each power/ground pin.
    - For standard cell, IR/EM view is identical to early view.
- IR power-grid view (Macros)
    - Power net is extracted from GDS with RC reduction performed.
    - Current taps are attached to the contact or user-specific via layer; for memory cell, some taps may be collapsed.
    - Decap value is characterized from spice simulation, and is attached to each current taps.
- EM power-grid view (Macros)
    - Power net is extracted from GDS without reduction.

- Current taps are attached to the contact or user-specific via layer; for memory cell, some taps may be collapsed.
- Decap value is characterized from spice simulation, and is attached to each current tap.
- Current is distributed based on propagation, current regions, or dynamic simulation.

There are two accuracy settings in rail analysis:

- XD (accelerated definition): used for early implementation stage IR/EM analysis
- HD (high definition): used for final verification stage IR/EM analysis

Based on accuracy settings and analysis, the power-grid views applied in rail analysis are as follows:

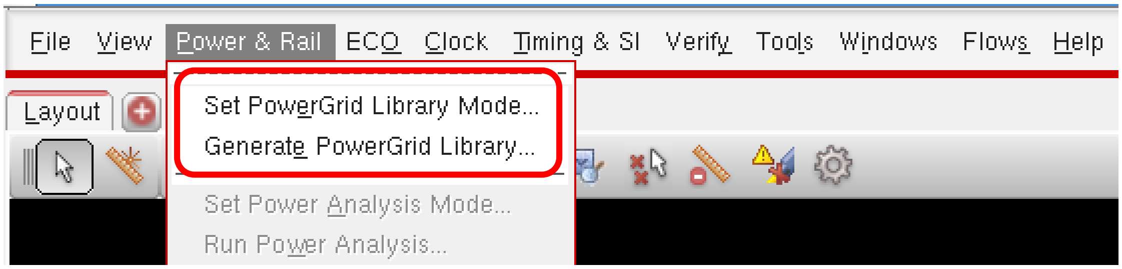
Analysis Type\Accuracy: Setting	**XD**	**HD**
Static IR	Early	IR
Static EM	Early	EM
Dynamic IR	Early	IR

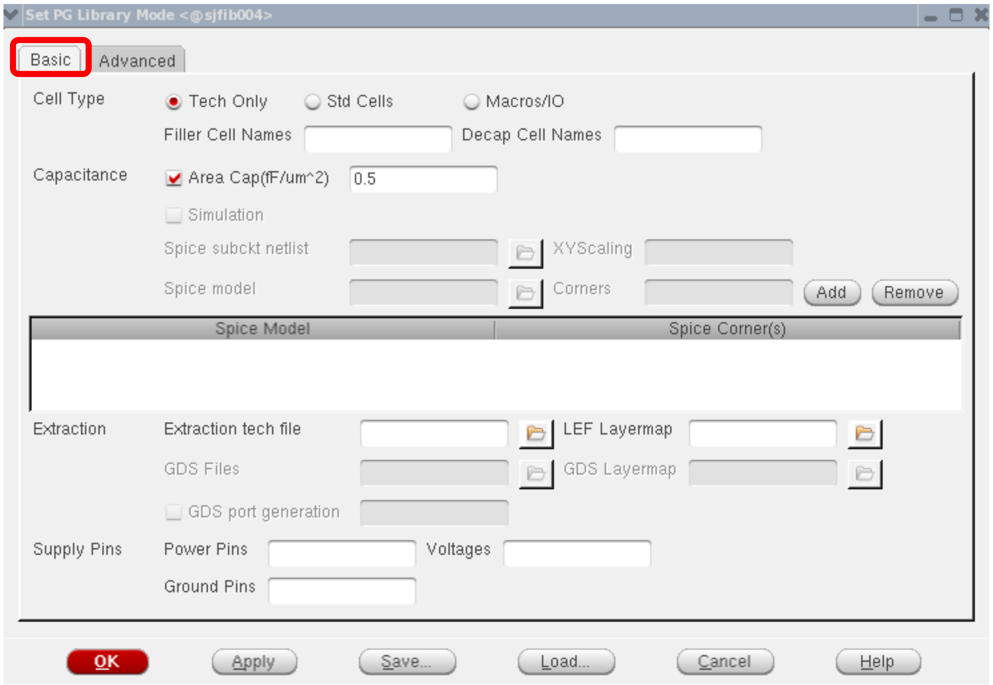
You use different accuracy settings at different stages of the design flow. For example, XD can be used for pipe-cleaning the design and in the early design phase, while HD can be used for the final verification. XD can also be used if performance is vital and detail macro analysis is not required. However, HD mode will produce more accurate results than XD.

So how do you create these different libraries for rail analysis?

There are three steps to running library generation:

1. Read the technology LEF and macro LEF.
2. Define the set_pg_library_mode switches. This can be done on the command line or through the GUI as shown in the following image).
3. Execute the generate_pg_library command. This can also be driven from the command line or the following GUI:





Set PG Library Mode <@sjfib004>

Basic | Advanced

Cell Type ● Tech Only ○ Std Cells ○ Macros/IO

Filler Cell Names: Decap Cell Names:

Capacitance ☑ Area Cap(fF/um^2): 0.5

□ Simulation

Spice subckt netlist: XYScaling:

Spice model: Corners: Add Remove

Spice Model	Spice Corner(s)

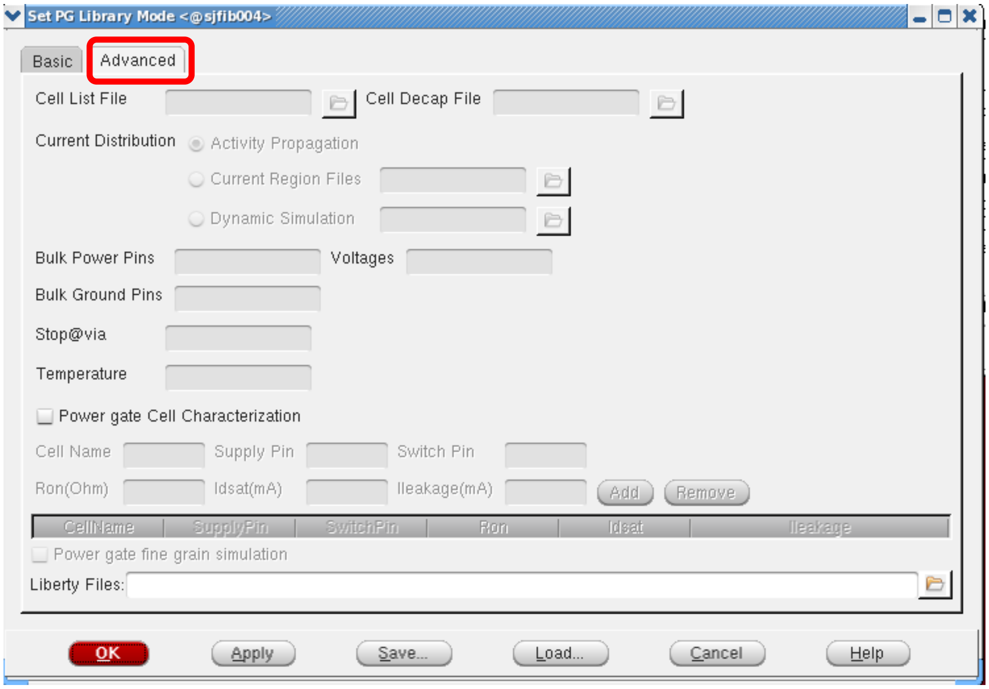
Extraction: Extraction tech file: LEF Layermap:

GDS Files: GDS Layermap:

□ GDS port generation

Supply Pins: Power Pins: Voltages:

Ground Pins:



Set PG Library Mode <@sjfib004>

Basic | Advanced

Cell List File: Cell Decap File:

Current Distribution ● Activity Propagation ○ Current Region Files ○ Dynamic Simulation

Bulk Power Pins: Voltages:

Bulk Ground Pins:

Stop@via:

Temperature:

□ Power gate Cell Characterization

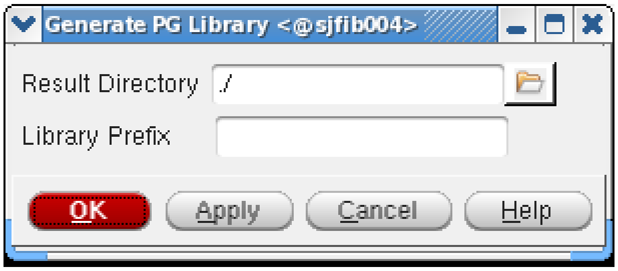
Cell Name: Supply Pin: Switch Pin:

Ron(Ohm): Idsat(mA): Ileakage(mA): Add Remove

CellName	SupplyPin	SwitchPin	Ron	Idsat	Ileakage

□ Power gate fine grain simulation

Liberty Files:



Generate PG Library <@sjfib004>

Result Directory: ./

Library Prefix:

The following three types of libraries will be generated:

1. Technology library
2. Standard cell power-grid library
3. Macro power-grid library

The steps below use the command line to enter the Voltus commands. Using the commands clearly demonstrates what is required for each type of model.

Generating Technology Library

Required input files are:

- technology LEF
- cell LEF
- QRC tech file
- lefdef layer map - optional

To generate a technology library:

1. Go to the power grid view generation work directory:

```
linux> cd pgv_gen
```

2. Delete any files and directories:

```
linux> rm –rf *
```

3. Start Voltus:

```
linux> voltus
```

Note: As an alternative to run all steps 4-7, run the following command:

```
source ../tcl/techonly.tcl
```

4. Read the LEF for all the cells to be characterized:

```
read_lib \
    -lef \
        ../data/lef/gsclib090_tech.lef \
        ../data/lef/gsclib090_macro.lef \
        ../data/lef/pso_header.lef \
        ../data/lef/pso_ring.lef \
        ../data/lef/pll.lef \
        ../data/lef/decap.lef
```

5. Configure Voltus for library characterization:

```
set_pg_library_mode \
  -celltype                techonly \
  -ground_pins             VSS \
  -power_pins              {VDD 0.9 VDDG 0.9 TVDD 0.9} \
  -decap_cells             {DECAP8 DECAP64 DECAP4 DECAP32 DECAP2 DECAP16 DECAP1} \
  -filler_cells            {FILL8  FILL64  FILL4  FILL32  FILL2  FILL16  FILL1} \
  -default_area_cap        0.01 \
  -cell_decap_file         ../data/voltus/decap.cmd \
  -extraction_tech_file    ../data/qrc/gpdk090_9l.tch \
  -lef_layermap            ../data/voltus/lefdef.layermap \
  -powergate_parameters { \
    {RING_SWITCH   TVDD VDD 750 0.5 4.0e-8} \
    {HEADER_SWITCH TVDD VDD 750 0.5 4.0e-8} \
  } \
```

What do these options mean?

cell_type	techonly – for technology library
power_pins	Identifies power pins and the corresponding voltage used for characterization
ground_pins	Identifies ground pins
extraction_tech_file	QRC tech file used during extraction
lef_layermap	Maps LEF layer names to the QRC tech file layer names, it is optional, without it Voltus will automatically generates one. It becomes optional.
filler_cells	List of filler cell
decap_cells	List of decap cells
default_area_cap	Defines cap value, which is area based (fF/um^2)
powergate_parameters	Defines power gate cells and parameters

Powergate_parameter definitions are the following:

```
-powergate_parameters {cell_name global_power_net switched_power_net Ron
Idsat Ileakage }
```

Ron — This is the “on” resistance of switch. It is the resistance across the power switch. The units are ohms.

Idsat — This is the saturation current through the switch. Units are mA.

Ileak — This indicates the leakage power of the switch. Units are mA.

Note: For technology library, user needs to provide above parameters for power gate cells, and the TECH view can be used for dynamic power analysis to generate power gate trigger file. However, the power-grid view is required for power up rail analysis.

6. Run library generation:

```
generate_pg_library -output tech_pgv
```

What do these options mean?

output — Specifies the name of the output directory in which power-grid libraries are created. The default is the current working directory.

It will result into the following summary:

```
** INFO:  (VOLTUS_LGEN-3606):
Power Grid View Generation Statistics:
  # Total number of cells: 503
  # TECH view created: 503 (100%)
```

What do you find in the output directory?

techonly.cl	Technology library
techonly.main.html	Html report

techonly.report A report for each cell's power/ground net containing:

- name of power and ground, voltage value for power pins

- capacitance value which is area-based

- number of taps

- user-defined Idsat, Ileakage, and Ron for POWERGATE cells

techonly.summary report containing:

-cell types: STDCELL, FILLER, DECOUPLING CAP, POWER_GATE

-pins, metal layers, number of current taps, etc.

-for more info, please refer to the User's Guide

Note: If you delete the characterization report after generating the libraries, a new summary report can be regenerated by:

```
check_power_library \
    -report_file   my.rpt \
                   tech_pgv/techonly.cl
```

Also, there are also options, `summary` and `–long_summary` to generate both the summary file and the long summary file.

7. Exit Voltus:

```
exit
```

Generating Standard Cells Power-grid Library

Now, let us generate a standard cells power-grid library. But include a spice netlist to ensure that Voltus can characterize the cell's natural coupling capacitance using spice simulation. The natural coupling capacitance will remove pessimism from the rail analysis. Note that PLL has been omitted because it is considered a macro.

Also, note that GDS has not been provided because standard cells are small and the IR drop due to their internal power grid is insignificant. If the solver considered this additional data, it will impact the rail analysis performance and memory usage. Even if you provide GDS, Voltus will not use it during standard cell characterization.

To create a PORT power grid views for standard cells:

1. Go to the power grid view generation work directory:

   ```
   linux> cd pgv_gen
   ```

2. Delete any files and directories:

   ```
   linux> rm –rf *
   ```

3. Start Voltus:

   ```
   linux> voltus
   ```

Note: As an alternative to running all steps 4-7, run the following command:

```
source ../tcl/std_cells.tcl
```

4. Read the LEF for all cells to be characterized. This is the same step used in generating technology library above; except we are not providing LEF for the PLL and low power cells.

   ```
   read_lib \
       -lef \
           ../data/lef/gsclib090_tech.lef \
           ../data/lef/buf_ao.lef \
           ../data/lef/decap.lef \
           ../data/lef/gsclib090_macro.lef \
           ../data/lef/pso_header.lef  \
           ../data/lef/pso_ring.lef \
           ../data/lef/pll.lef
   ```

5. Configure Voltus for library characterization. This is very similar to step used in generating technology library as before, but with following exceptions:

    a. Changing the celltype to stdcells
    b. Adding spice netlist and model files

```
set_pg_library_mode \
    -ground_pins                VSS \
    -power_pins                 {VDD 0.9 TVDD 0.9} \
    -decap_cells               {DECAP8 DECAP64 DECAP4 DECAP32 DECAP2 DECAP16 DECAP1}\
    -filler_cells              { FILL8  FILL64  FILL4  FILL32  FILL2  FILL16  FILL1}\
    -celltype                   stdcells \
    -cell_decap_file            ../data/voltus/decap.cmd \
    -cell_list_file             ../data/voltus/cell.list \
    -spice_subckts { \
        ../data/netlists/gsclib090.sp \
        ../data/netlists/pso_header.spi \
        ../data/netlists/pso_ring.spi \
    } \
    -lef_layermap               ../data/voltus/lefdef.layermap \
    -current_distribution       propagation \
    -spice_models               ../data/netlists/spectre_load.sp \
    -extraction_tech_file       ../data/qrc/gpdk090_9l.tch \
    -powergate_parameters { \
        {RING_SWITCH   TVDD VDD} \
        {HEADER_SWITCH TVDD VDD} \
    }
```

celltype   Defines standard cell type of power-grid library

spice_models   Spectre file that points to the spectre models. You can also use spice models.

spice_subckts   Standard cells spice netlists with parasitic on the signal nets only. Do not extract the parasitic of the p/g rails.

powergate_parameters   Define power gate cells: cell name, global power pin name and switched power pin name

6. Now run library characterization:

```
generate_pg_library -output stdcell_pgv
```

Note: In Voltus 14.1, there are warnings indicating missing spice netlist for a list of cells. This is acceptable because these cells have missing spice netlist. In the latest release, you need to define a cell list with cell names that have netlist or Voltus will error out.

It will take approximately three minutes to run. When it is done, you see the following summary message:

** INFO: (VOLTUS_LGEN-3265):

Power Grid View Generation Statistics:

# Total number of cells: 488
# EARLY view created: 488 (100%)
# IR view created: 488 (100%)
# EM view created: 488 (100%

What do you find in the output directory?

stdcells.cl	Standard cell library
stdcells.main.html	HTML report
stdcells.report	A report containing for each cell's power/ground net: - name of power and ground, voltage value for power pins - capacitance value which is simulation based - number of taps - Idsat, Ileakage and Ron for POWERGATE cells calculated based on simulation - detailed information for power gate cells
stdcells.summary	Report containing: - cell types: STDCELL, FILLER, DECOUPLING CAP, POWER_GATE - pins, metal layers, number of current taps, etc. - for more info, please refer to the User's Guide

Because the spice netlist is provided for power-gate cells, you will see the detailed report about power-gate cells in stdcell_pgv/stdcells.report:

```
-------------------------------------------------------------------------------------------------
HEADER_SWITCH
                    TVDD(0.9000)     1.1e-15          EARLY(2 taps) EM(2 taps) IR(2 taps)
                    VDD(0.9000)     2.3e-15          EARLY(18 taps) EM(18 taps) IR(18 taps)
                    VSS(0.0000)     3.2e-15          EARLY(14 taps) EM(14 taps) IR(14 taps)
INFO: This is a POWERGATE cell. The EM view of this cell.
Following Powergates are driven by enable Pin NSLEEPIN
……
Following Powergates are driven by enable Pin NSLEEPIN
        Powergate PG_NSLEEPIN_1 has Idsat = 0.0011678A, Ileakage = 7.204e-05A, Ron = 127.361 Ohm at Voltage = 0.45
                                    Idsat = 0.0017954A, Ileakage = 0.00010432A, Ron = 94.5048 Ohm at Voltage = 0.63
                                    Idsat = 0.002416A, Ileakage = 0.00014302A, Ron = 79.3986 Ohm at Voltage = 0.81
                                    Idsat = 0.002722A, Ileakage = 0.00016488A, Ron = 74.7294 Ohm at Voltage = 0.9
                                    Idsat = 0.003328A, Ileakage = 0.000215A, Ron = 68.5897 Ohm at Voltage = 1.08
                                    Idsat = 0.003946A, Ileakage = 0.0002786A, Ron = 65.1225 Ohm at Voltage = 1.26
                                        Powergate PG_NSLEEPIN_1 has Roff = 0 Ohm at Voltage = 0.45
                                                                    Roff = 0 Ohm at Voltage = 0.63
```

```
                                                                 Roff = 0 Ohm at Voltage = 0.81
                                                                 Roff = 0 Ohm at Voltage = 0.9
                                                                 Roff = 0 Ohm at Voltage = 1.08
                                                                 Roff = 0 Ohm at Voltage = 1.26
INFO: Powergate Ids-Vds tables are present.

 Cell Type = POWER_GATE                                     Cell_Status: PASS
--------------------------------------------------------------------------------------------------------------------
```

7. Exit Voltus:

```
exit
```

Generating a Macros Power-grid Library

To create the most accurate power grid view, you need LEF, GDS and Spice. The LEF provides the port locations of the macro, the GDS provides the internal rail structure of the macro, and the spice allows you to accurately compute and attach the tap current sources within the macro. The Voltus simulates the spice netlist and determines the relative power consumed by each device in the netlist. If the spice netlist contains XY coordinates for each device, the tap current source will be attached to the appropriate location in the model.

This allows Voltus to accurately model current consumption within the macro and to compute IR drop on rails internal to the macro.

The GDS layer map file you will use will have the following format:

```
<diff|via|poly|metal> <QRC Layer name> gds <number> <purpose>
```

The purpose is optional if it is the default value of 0. Here's an example:

```
diff        diffusion       gds   5
metal       METAL_1         gds   7
metal       METAL_2         gds   9
metal       METAL_3         gds   11
poly        POLYCIDE        gds   3
via         CONT            gds   6
via         VIA_1           gds   8
via         VIA_2           gds   10
```

There are two important points that need to be highlighted:

- The layermap file only supports one via between layers. However, it is possible that your QRC tech file has two contacts going down from M1: one connects M1 to diffusion, while the other connects M1 to poly. If this is the case, use the contact that connects M1 to diffusion because any power grid connection to poly is used to tie off a cell input and will not affect the current draw on the grid. However, M1 connected to diffusion is an integral part of the power rail. This is from where the transistors in the cells get their power.

- The second point is regarding the layer map file. All layers from the contact up must match the names in the QRC tech file. The diff layer (called diffusion in the example and highlighted in red) MUST NOT MATCH a layer in the QRC tech file because the diffusion layer is derived by the extractor. If we give it a name in the QRC tech file, the derived layer will overwrite the input layer. To map the GDS layer, select a name of a layer used to define p-diffusion in the QRC tech file.

Creating a detailed power grid views for a macro cell

1. Go to the power grid view generation work directory:

```
linux> cd pgv
```

2. Delete any files and directories:

```
linux> rm -rf *
```

3. Start Voltus:

```
linux> voltus
```

Note: As an alternative to running all steps 4-7, run the following command:

```
source ../tcl/pll_macro.tcl
```

4. Read the LEF for all cells to be characterized. This is the similar to characterizing the PORT power grid view:

```
read_lib \
    -lef \
        ../data/lef/gsclib090_tech.lef \
        ../data/lef/pll.lef
```

5. Configure Voltus for library characterization:

```
set_pg_library_mode \
      -gds_files              ../data/gds/pll.gds \
      -ground_pins            VSS \
      -cell_list_file         ../data/voltus/macro.list \
      -power_pins             {VDD 0.9} \
      -celltype               macros \
      -spice_subckts          ../data/netlists/pll.sp \
      -gds_layermap           ../data/voltus/gds.layermap \
      -lef_layermap           ../data/voltus/lefdef.layermap \
      -stop@via               CONT \
      -spice_models          ../data/netlists/spectre_load.sp \
      -current_distribution  propagation \
      -extraction_tech_file  ../data/qrc/gpdk090_9l.tch
```

The steps for macros of a power grid view's library characterization are more complicated than that of standard cells. What needs to change?

celltype

Specify as macros

gds_files

The macro's GDS file. It is used to extract the internal power rails of the cell

gds_layermap

Maps the QRC tech file layers to the layer numbers in the GDS file

spice_models

Spectre file that points to the spectre models. You can also use spice models

spice_subckts

The macro's spice or CDL netlist. If the macro is large, use CDL. If the macro is small, use a spice netlist and include parasitic on the nets. Never extract power/ground rail parasitics

For the most accurate results, include the XY location of the transistors in the netlist. If these are included, Voltus will place a tap current source at this location. The current source will draw the current based on a spice simulation of the circuit.

If the netlist does not contain XY coordinates, Voltus will still extract and apply the tap current sources at the location of the stop vias. However, the current will be equally distributed among all the tap current sources.

stop@via

This tells the extractor to stop extracting once it hits the specified via. Wherever this via is located, Voltus will search for the nearest transistor from the spice netlist and place a tap current source at this location. It is most accurate to use the contact connected to diffusion (not poly). VIA1 has also proven to be accurate. If there's a conflict between the via names in the QRC tech file and the tech LEF, use the QRC tech file name.

current_distribution	propagation – This is Voltus library simulator to determine the propagation activity to distribute the current.

6. Run power-grid library generation:

```
generate_pg_library -output macro_pgv/
```

When it is done, you will see a summary message:

```
** INFO:  (VOLTUS_LGEN-3265):
Power Grid View Generation Statistics:
        # Total number of cells: 1
        # EARLY view created: 1 (100%)
        # IR view created: 1 (100%)
        # EM view created: 1 (100%)
```

Note: The following warning can be ignored (It occurs due to the ratio between the data size defined in GDS and the data size defined in LEF file):

```
** WARN:  (VOLTUS_LGEN-3591): Coordinates in spice netlist do not match
the
coordinates in the layout. This may result in improper current taps
creation. Check the coordinates in spice netlist and set
-spice_subckts_xyscaling option in set_pg_library_mode command.
CELL RATIO= 11.000000
** WARN:  (VOLTUS_LGEN-3591): Coordinates in spice netlist do not match
the
coordinates in the layout. This may result in improper current taps
creation. Check the coordinates in spice netlist and set
-spice_subckts_xyscaling option in set_pg_library_mode command.
CELL RATIO= 11.000000
```

Note: To see more details about the job, you can add following to the run script:

```
set_advanced_pg_library_mode -verbosity true
```

With the above option set, there are more runtime data and log files generated. You can check extraction information in macro_pgv/macros_pll_temp /pll_SHC_000.log and look for the following to ensure the extractor actually extracted down to stop@via:

```
** INFO:  (VOLTUS_EXTR-1254): ... ExtractCellViaR for cell pll_SHC, layer
CONT (METAL_1, SOURCE_DRAIN).

** INFO:  (VOLTUS_EXTR-1502): Cluster distance and size in layer CONT:
20000 : 1000
```

```
** INFO:  (VOLTUS_EXTR-1501): Number of vias in this cut layer: 2283

** INFO:  (VOLTUS_EXTR-1500): Number of vias clustered in this cut layer:
2219

** INFO:  (VOLTUS_EXTR-1509): Via Cluster Statistics for layer CONT.
Number of cluster types 3 and its list [Cuts in cluster, Num of clusters]
[ 1, 29 ]  [ 2, 27 ]  [ 275, 8 ]
```

7. Exit Voltus:

```
exit
```

Loading the Design

Start Voltus:

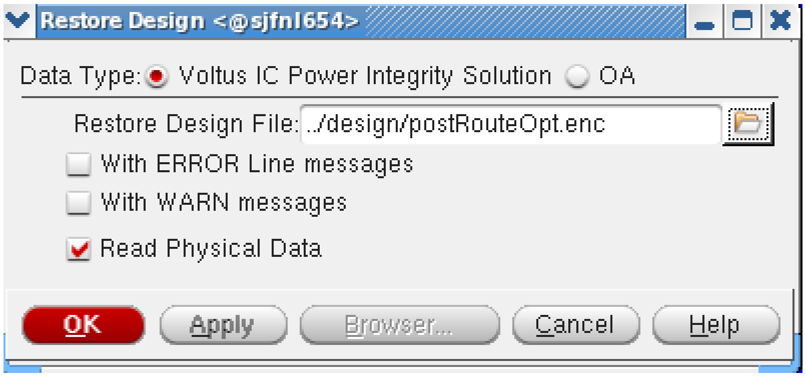
```
cd static
voltus
```

Load Design: You can load a design into Voltus in one of following two ways:

1. It can restore an EDI database:
   a. Using the GUI

   From the GUI:

   Select, Design → Read Design. The following window will appear:



Restore Design <@sjfnl654>

Data Type: ● Voltus IC Power Integrity Solution ○ OA

Restore Design File: ../design/postRouteOpt.enc

☐ With ERROR Line messages

☐ With WARN messages

☑ Read Physical Data

   b. Using the command line

   From the command line, run:

```
read_design       -physical_data       ../design/postRouteOpt.enc.dat
super_filter
```

2. It can load libraries and netlists etc. incrementally:

   a. Alternatively, you can read the design piece-meal by loading libraries and netlists etc. with commands such as read_lib, read_verilog, etc.:

```
set lefs [list]
lappend lefs ../data/lef/gsclib090_tech.lef
lappend lefs ../data/lef/gsclib090_macro.lef
lappend lefs ../data/lef/pll.lef
lappend lefs ../data/lef/decap.lef
lappend lefs ../data/lef/pso_header.lef
lappend lefs ../data/lef/pso_ring.lef
lappend lefs ../data/lef/buf_ao.lef
```

```
read_lib -lef $lefs

read_view_definition ../design/viewDefinition.tcl

read_verilog ../design/postRouteOpt.enc.dat/super_filter.v.gz
set_top_module super_filter -ignore_undefined_cell

read_def ../design/super_filter.def.gz
```

b. Load CPF file: Once the design is loaded, you can read CPF file for the power domain information:

```
read_power_domain -cpf ../design/super_filter.cpf
```

Or
Using the following options to define each power/ground net's voltage:

```
set_dc_sources VDD_AO       -power -voltage 0.9 -force
set_dc_sources VDD_external -power -voltage 0.9 -force
set_dc_sources VDD_ring     -power -voltage 0.9 -force
set_dc_sources VDD_column   -power -voltage 0.9 -force
set_dc_sources VSS          -ground
```

c. Load SPEF file: Now load the corresponding SPEF:

```
read_spef \
      -rc_corner  RC_wc_125   \
      -decoupled              \
                  ../design/postRouteOpt_RC_wc_125.spef.gz
```

Static Power Analysis

Now, let us prepare for static power analysis. You can enter the following commands into the Voltus console. Alternatively, the same commands are found in the file ../tcl/static.tcl. You can cut/paste the command from the file into the Voltus console.

1. Define the power analysis mode:

```
set_power_analysis_mode -reset
set_power_analysis_mode \
  -analysis_view          AV_wc_on \
  -write_static_currents   true \
  -binary_db_name          staticPower.db \
  -create_binary_db        true \
  -method                  static
```

Why are we setting the variables to these values?

analysis_view
Voltus can only analyze power for one view at a time. Here, we define the view to be analyzed.

write_static_currents
This writes each instance's current waveform to a binary database. For static power, this will be a single value (average). But for dynamic power analysis, a current waveform will be saved.

create_binary_db
This specifies if the binary power db will be written. This db contains instances, locations and power. It allows for cross probing between the power debugger and the layout.

binary_db_name
This is the name of the binary power database. This is not to be confused with the current file.

Method
This specifies the type of analysis to be performed: static, dynamic_vectorless or dynamic_based.

How is the power computed for an instance? Internal power of the instance is derived from the .lib's power look up table; leakage power of the instance is calculated based on .lib; switching power is calculated based on charging and discharging output load capacitance. You can explicitly define the power of an instance or cell with the set_power command. Here is its format:

```
set_power -type {cell|inst} <instname> -pg_net <net> <power number>
```

In the example below, first one applies the power to the instance, whereas the other applies it to all PLL instances.

```
set_power -reset
set_power 0.5 –type cell pll -pin VDD
set_power 0.5 –type inst u0  –pin VDD
```

Power grid views can be used for power analysis when you don't specify .libs with power LUT or you don't define a power value with set_power. You use the set_power_analysis_mode – power_grid_library option to do this.

2. Controlling toggle rates:

Voltus will use the create_clock and create_generated_clock from the SDC file to control the toggle rates on clocks. However, SDCs do not define switching activity for primary inputs and other nets. SDCs also do not define how often the output of a clock gate toggles.

If available, these toggle rates can be defined through a toggle count file (TCF) or a value change dump file (VCD). VCD will be covered later.

To control the activity rates on the specific instances, nets and primary inputs use the following command:

```
set_switching_activity
```

For example, to control the activity on all primary inputs:

```
set_switching_activity -reset
set_switching_activity -input_port rst -activity 0.25 -duty 0.3
```

The ‘`propagate_activity`’ command propagates the activity file in the database after it is read in using the `read_activity_file` command. This command propagates the activity for all primary inputs, nets, and other devices in the design where the activity has not been previously defined through user attributes. The `propagate_activity` command can be used to propagate activities in the design and check for activity annotation before running power calculation. This is not a mandatory step as the execution of `report_power` will also do the same, and then do power calculation based on the propagated activities.

```
 propagate_activity
```

To verify the activity was set properly, run:

```
get_activity –port rst
```

The following will be reported:

```
rst user_defined_activity 0.3 3.125e+07
```

The report indicates:

- Port name
- The activity defined by the user (as opposed to other sources)
- The duty cycle
- The number of toggles per second

But how is 3.125e+07 related to activity of 0.25 we defined earlier?

The 0.25 means the port will toggle every 4 clock cycles. Because the port is clocked by an 8ns clock, the port will toggle every 32ns. This translates to 3.125e+07 toggles per second:

0.25 * 1 / 8e-09 = 3.125e+07

Often, it is useful to control the activity of all clock gating instances. The clock gating cells in this design have the RC_CGIC prefix. Set the toggle rate of the clock gate outputs to 0.1:

```
set_switching_activity \
      -pin [get_pins -of_objects [get_cells RC_CGIC* -hierarchical] \
      -filter "@direction == out"] \
      -activity 0.1
propagate_activity
```

3. Now, validate that the activities were properly applied:

```
get_activity \
      -pin [get_pins -of_objects [get_cells RC_CGIC* -hierarchical] \
      -filter "@direction == out"]
```

You can use set_default_switching_activity to set default values on unassigned nets:

```
set_default_switching_activity \
      -input_activity           0.3 \
      -period                   4.0 \
      -clock_gates_output_ratio  0.5
```

Note: You can specify the directory where the power results will be stored. This is useful, so we do not clutter our work directory.

```
set_power_output_dir staticPowerResults
```

4. Finally, you can execute report_power to run the analysis. You can specify the –outfile <> option to create a text report with power values for each instance:

```
report_power -outfile static.rpt
```

In the staticPowerResults directory, you see:

staticPower.db	This is creation and name determined by the set_power_mode command. It is used for analyzing power plots. It cross references instance names and locations with the results.
staticPower.db.cnstr.tcl	This stores the commands used to setup the power analysis.
static.rpt	This is the Power report (as specified with the report_power command)
static_<net>.ptiavg	This is the current file for each power rail

5. Review the staticPowerResults/static.rpt file. Note the report summarizes power consumption, by groups (sequential cells, macros, clocks, etc.), power net and clock domain.

   You can experiment with different report formats by modifying the report_power command. For example, if you want to report the power of a specific instance, you can run:

```
report_power –instances {u0} –outfile u0.rpt
```

   For all instances, you can run:

```
report_power –instances {*} –outfile all.rpt
```

   Or to report on a particular level of hierarchy:

```
report_power –instances {ring} –outfile ring.rpt
```

   The text reports are not exciting to read. A graphical representation will be ideal. Voltus offers multiple ways to visualize power consumption in the design: Power Debugger and the Power Results Viewer.

   But, before we move on, remember how you queried the activity rate of various pins with the get_activity command? After running report_power, you can run the get_power command to get activity rates, clocks, duty cycles and more:

```
get_power -nets rst -attribute {ref_clock toggle_rate duty_cycle}
```

Power Debugger

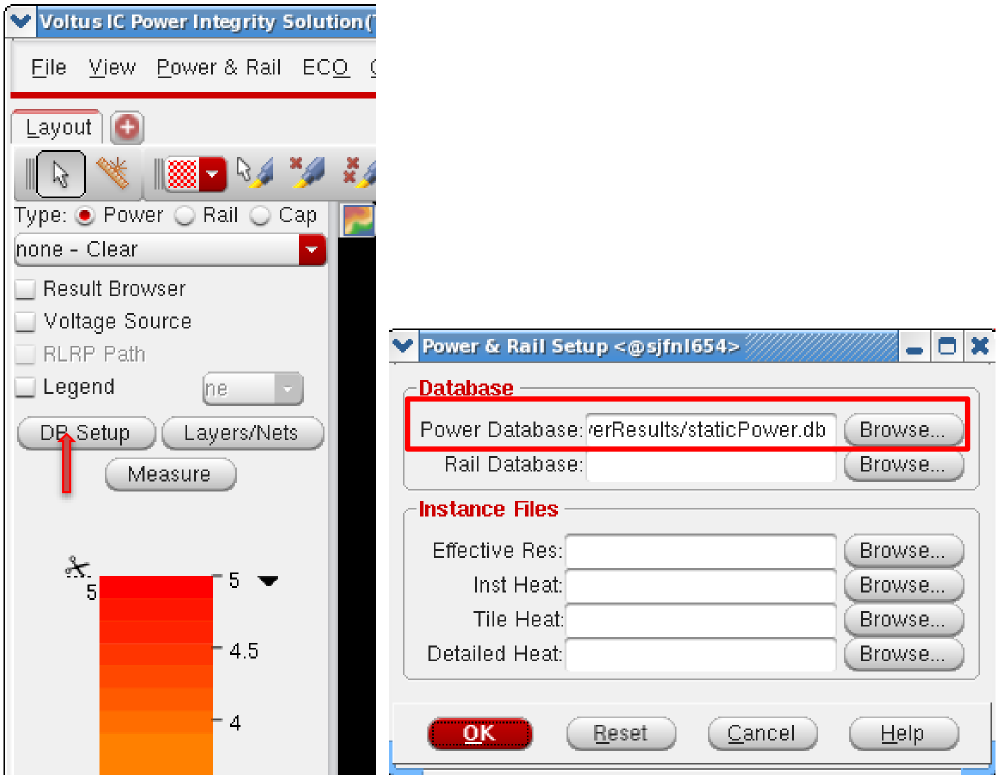
1. Enter start_gui to bring the Voltus GUI up.

2. Load power data:

   a. The power is automatically loaded after running static power analysis
   b. Use TCL to load the power data:

      ```
      read_power_rail_results \
      -power_db
      staticPowerResults/staticPower.db
      ```

   c. Optional: Load through Voltus GUI: select Power & Rail -> Power & Rail Plot to bring the Voltus debug panel up; Click DB Setup button to bring Power & Rail Setup form up; fill in Power Database with “staticPowerResults/staticPower.db”:



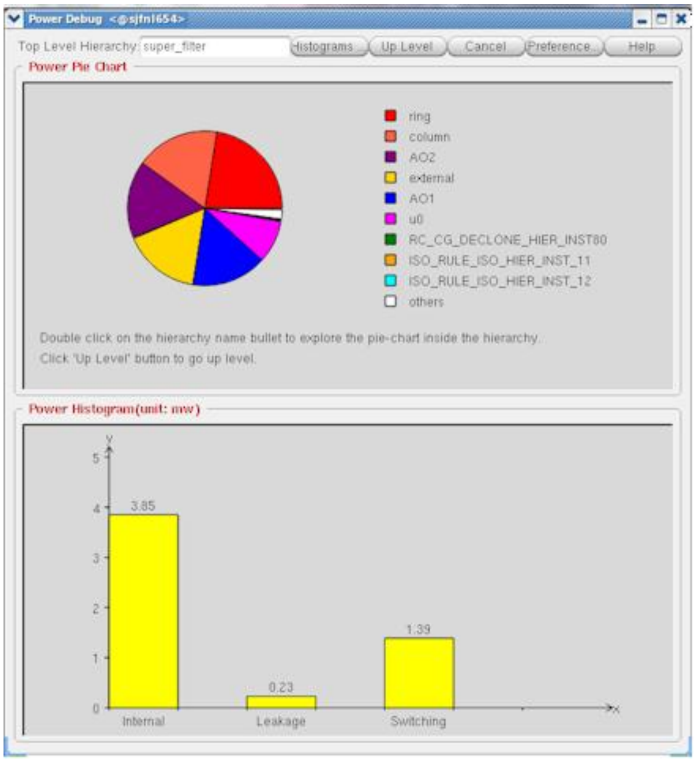
Type: ● Power ○ Rail ○ Cap

none - Clear

□ Result Browser
□ Voltage Source
□ RLRP Path
□ Legend

Power Database: erResults/staticPower.db
Rail Database:
Effective Res:
Inst Heat:
Tile Heat:
Detailed Heat:

3. The Power Debugger is invoked from the GUI. Select Power & Rail-> Histograms. This will display a window that resembles the following:



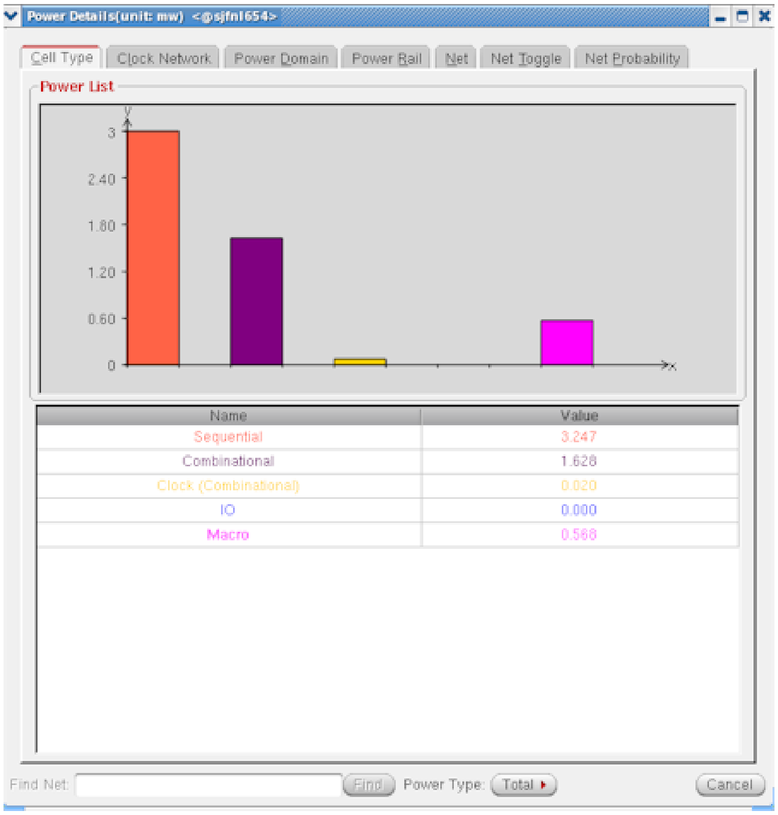
Double click on the hierarchy name bullet to explore the pie-chart inside the hierarchy.
Click 'Up Level' button to go up level.

The pie chart describes how the power is distributed between the levels of hierarchy. Double-click a pie slice, and you will drop into that level of hierarchy.

Note the bottom of the window displays a histogram showing how the power is distributed between internal, leakage and switching power for the current level of hierarchy. The values will change as you click the pie slides to go into a lower level of hierarchy.

4. Click the “Up Level” button to go back to the top level of hierarchy.

5. Back at the top level of hierarchy, click the “Histograms” button. A new window will appear that looks like:

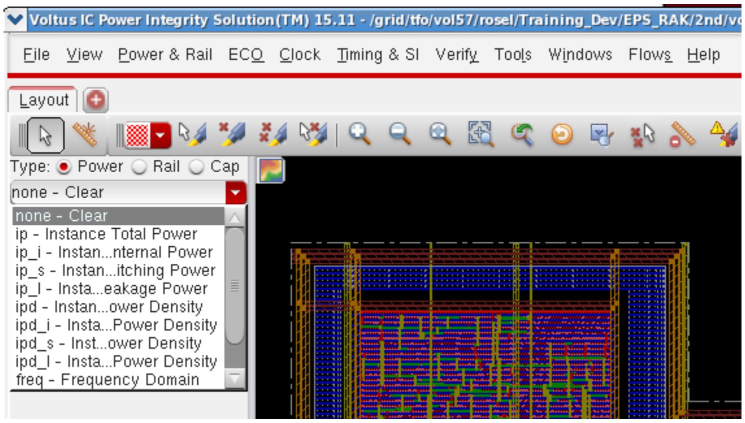


Name	Value
Sequential	3.247
Combinational	1.628
Clock (Combinational)	0.020
IO	0.000
Macro	0.568

The window contains various tabs that help visualize how the power is distributed and why:

Cell Type	Shows how power is distributed among sequential, standard cells, IOs and macros
Clock Network	Power by clock domain
Power Domain	Power by power domain
Power Rail	Power by power net
Net	Nets with the highest power
Net Toggle	Nets with the highest toggle rate (toggles/second)
Net Probability	Nets with the highest toggle probability

6. An alternative way to visualize the power results is to use the power viewer. In the Voltus debug panel, select Power and select a type of plots you want to plot:


Voltus IC Power Integrity Solution(TM) 15.11 - /grid/tfo/vol57/rosel/Training_Dev/EPS_RAK/2nd/vo
File View Power & Rail ECO Clock Timing & SI Verify Tools Windows Flows Help
Layout
none - Clear
none - Clear
ip - Instance Total Power
ip_i - Instan...nternal Power
ip_s - Instan...itching Power
ip_l - Insta...eakage Power
ipd - Instan...ower Density
ipd_i - Insta...Power Density
ipd_s - Inst...ower Density
ipd_l - Insta...Power Density
freq - Frequency Domain

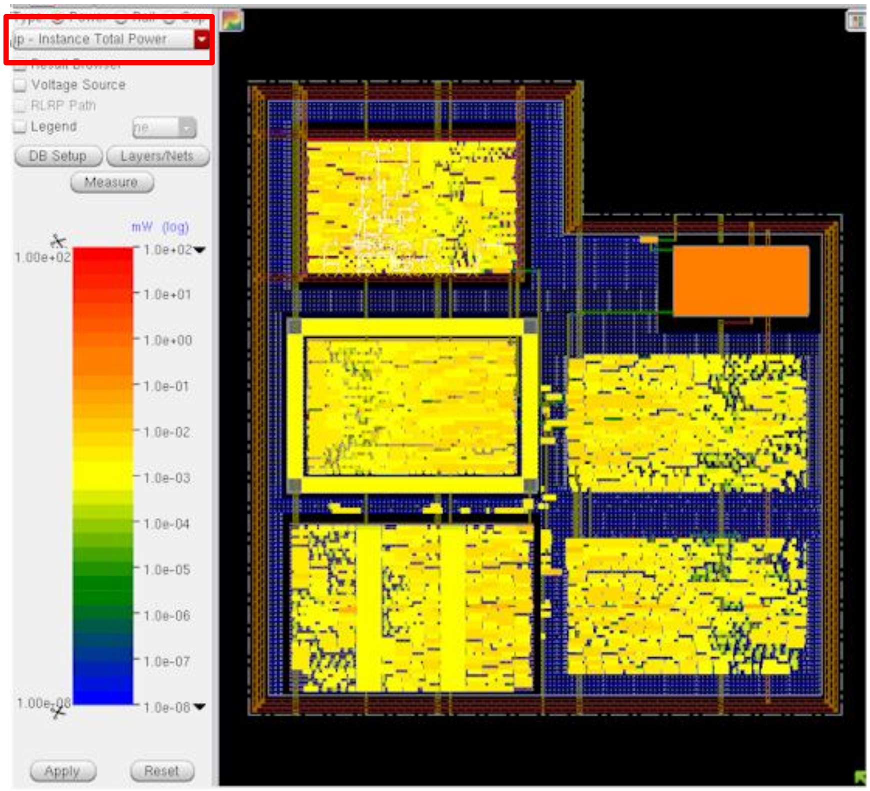
Type: ● Power ○ Rail ○ Cap

If you look at the layout view, each instance will be colored based on the data value. Alternatively, you can look at a histogram for all instances in the design. The power viewer plots a color map of the following values for each instance:

Key word	**Description**
ip	Total instance power
ip_i	Internal instance power
ip_s	Switching instance power
ip_l	Leakage instance power
ipd_t	Total instance power density
ipd_i	Internal instance power density
ipd_s	Switching instance power density
ipd_l	Leakage instance power density
freq	Dominant frequency of each instance
slack	WNS of the path containing that instance
td	Transition density; transitions/second=activity * freq

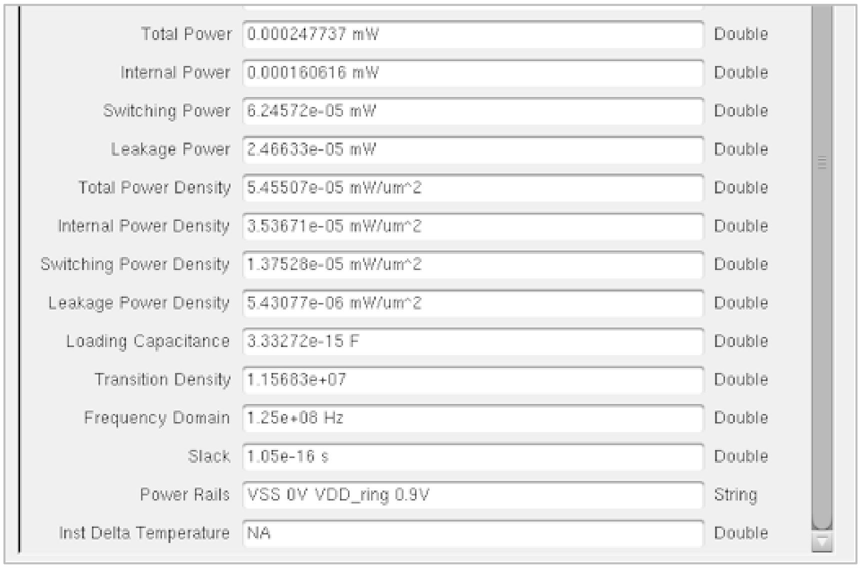
The GUI helps you view results of rail and power analyses. Hence, the GUI tends to ask for more information than is necessary at any one time.

Let’s look at the “*ip*” (instance power): Select *ip* on the left side pull down menu. The *ip* plot will be overlaid on the design.



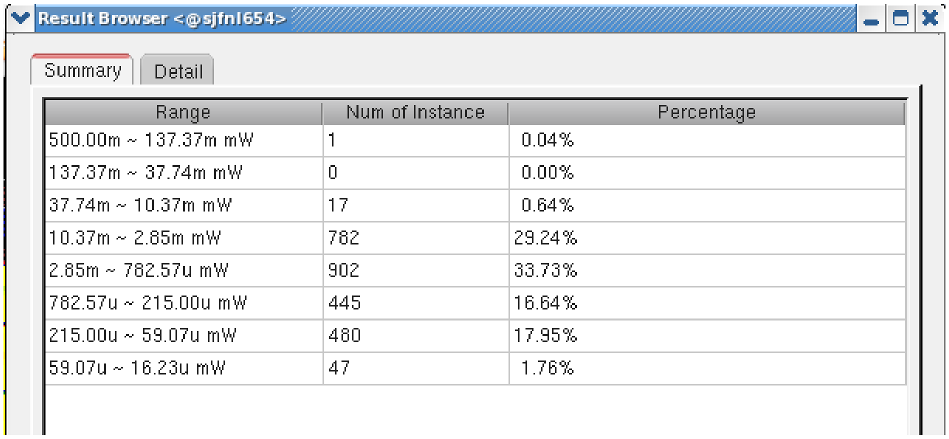
Let us look at each instance's total power distribution because this might identify areas of the design that have high current, which could lead to larger than expected IR drop.

7. Check instance power: select an instance and press "*q*" which will give you all the power related information on *Attribute Viewer*.



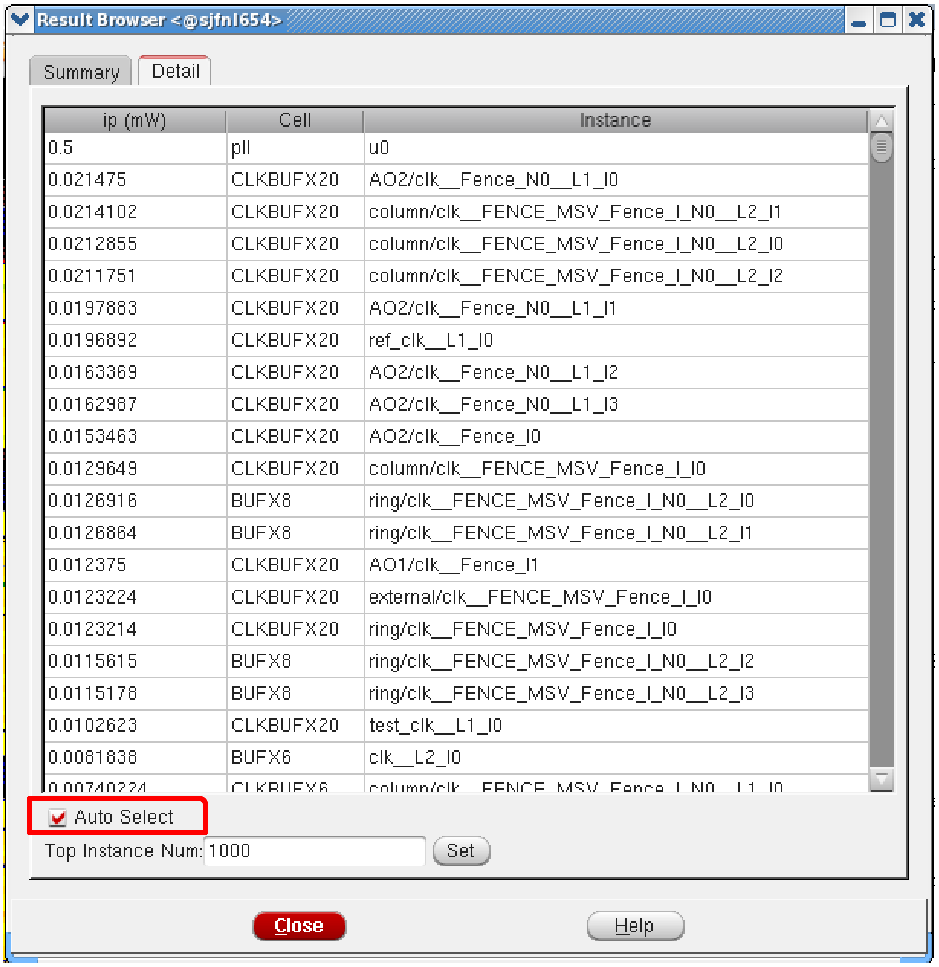
Total Power	0.000247737 mW	Double
Internal Power	0.000160616 mW	Double
Switching Power	6.24572e-05 mW	Double
Leakage Power	2.46633e-05 mW	Double
Total Power Density	5.45507e-05 mW/um^2	Double
Internal Power Density	3.53671e-05 mW/um^2	Double
Switching Power Density	1.37528e-05 mW/um^2	Double
Leakage Power Density	5.43077e-06 mW/um^2	Double
Loading Capacitance	3.33272e-15 F	Double
Transition Density	1.15683e+07	Double
Frequency Domain	1.25e+08 Hz	Double
Slack	1.05e-16 s	Double
Power Rails	VSS 0V VDD_ring 0.9V	String
Inst Delta Temperature	NA	Double

8. Results Browser: Check *Result Browser* button on the left side debug panel, the following Result Browser form will appear with the information displayed in Summary tab:



Range	Num of Instance	Percentage
500.00m ~ 137.37m mW	1	0.04%
137.37m ~ 37.74m mW	0	0.00%
37.74m ~ 10.37m mW	17	0.64%
10.37m ~ 2.85m mW	782	29.24%
2.85m ~ 782.57u mW	902	33.73%
782.57u ~ 215.00u mW	445	16.64%
215.00u ~ 59.07u mW	480	17.95%
59.07u ~ 16.23u mW	47	1.76%

Select the Detail tab. You will see the following, which lists the instances from the highest total instance power to the lowest instance power:

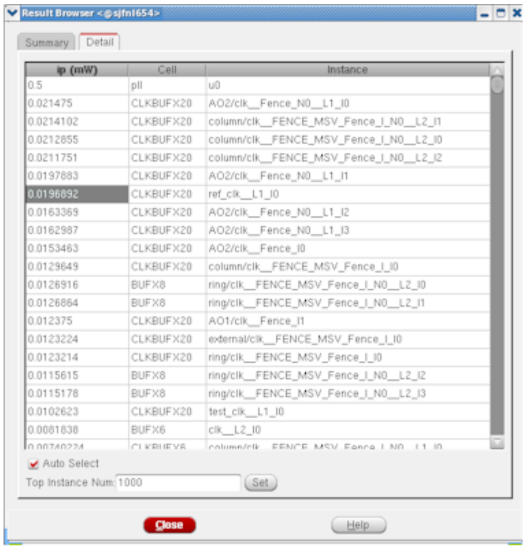


ip (mW)	Cell	Instance
0.5	pll	u0
0.021475	CLKBUFX20	AO2/clk__Fence_N0__L1_I0
0.0214102	CLKBUFX20	column/clk__FENCE_MSV_Fence_I_N0__L2_I1
0.0212855	CLKBUFX20	column/clk__FENCE_MSV_Fence_I_N0__L2_I0
0.0211751	CLKBUFX20	column/clk__FENCE_MSV_Fence_I_N0__L2_I2
0.0197883	CLKBUFX20	AO2/clk__Fence_N0__L1_I1
0.0196892	CLKBUFX20	ref_clk__L1_I0
0.0163369	CLKBUFX20	AO2/clk__Fence_N0__L1_I2
0.0162987	CLKBUFX20	AO2/clk__Fence_N0__L1_I3
0.0153463	CLKBUFX20	AO2/clk__Fence_I0
0.0129649	CLKBUFX20	column/clk__FENCE_MSV_Fence_I_I0
0.0126916	BUFX8	ring/clk__FENCE_MSV_Fence_I_N0__L2_I0
0.0126864	BUFX8	ring/clk__FENCE_MSV_Fence_I_N0__L2_I1
0.012375	CLKBUFX20	AO1/clk__Fence_I1
0.0123224	CLKBUFX20	external/clk__FENCE_MSV_Fence_I_I0
0.0123214	CLKBUFX20	ring/clk__FENCE_MSV_Fence_I_I0
0.0115615	BUFX8	ring/clk__FENCE_MSV_Fence_I_N0__L2_I2
0.0115178	BUFX8	ring/clk__FENCE_MSV_Fence_I_N0__L2_I3
0.0102623	CLKBUFX20	test_clk__L1_I0
0.0081838	BUFX6	clk__L2_I0
0.00740224	CLKBUFX6	column/clk__FENCE_MSV_Fence_I_N0__L1_I0

☑ Auto Select

Top Instance Num: 1000 Set

Auto select: you can highlight any instance on the list and the instance will be auto-zoomed.



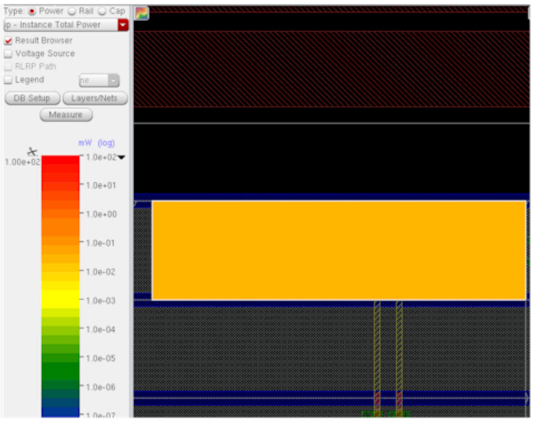
Result Browser <@sjfnl654>

Summary | Detail

ip (mW)	Cell	Instance
0.5	pll	u0
0.021475	CLKBUFX20	AO2/clk__Fence_N0__L1_I0
0.0214102	CLKBUFX20	column/clk__FENCE_MSV_Fence_I_N0__L2_I1
0.0212855	CLKBUFX20	column/clk__FENCE_MSV_Fence_I_N0__L2_I0
0.0211751	CLKBUFX20	column/clk__FENCE_MSV_Fence_I_N0__L2_I2
0.0197883	CLKBUFX20	AO2/clk__Fence_N0__L1_I1
0.0196892	CLKBUFX20	ref_clk__L1_I0
0.0163369	CLKBUFX20	AO2/clk__Fence_N0__L1_I2
0.0162987	CLKBUFX20	AO2/clk__Fence_N0__L1_I3
0.0153463	CLKBUFX20	AO2/clk__Fence_I0
0.0129649	CLKBUFX20	column/clk__FENCE_MSV_Fence_I_I0
0.0126916	BUFX8	ring/clk__FENCE_MSV_Fence_I_N0__L2_I0
0.0126864	BUFX8	ring/clk__FENCE_MSV_Fence_I_N0__L2_I1
0.012375	CLKBUFX20	AO1/clk__Fence_I1
0.0123224	CLKBUFX20	external/clk__FENCE_MSV_Fence_I_I0
0.0123214	CLKBUFX20	ring/clk__FENCE_MSV_Fence_I_I0
0.0115615	BUFX8	ring/clk__FENCE_MSV_Fence_I_N0__L2_I2
0.0115178	BUFX8	ring/clk__FENCE_MSV_Fence_I_N0__L2_I3
0.0102623	CLKBUFX20	test_clk__L1_I0
0.0081838	BUFX6	clk__L2_I0
0.00740224	CLKBUFX6	column/clk__FENCE_MSV_Fence_I_N0__L1_I0

☑ Auto Select

Top Instance Num: 1000 [Set]



Type: ● Power ○ Rail ○ Cap

ip - Instance Total Power

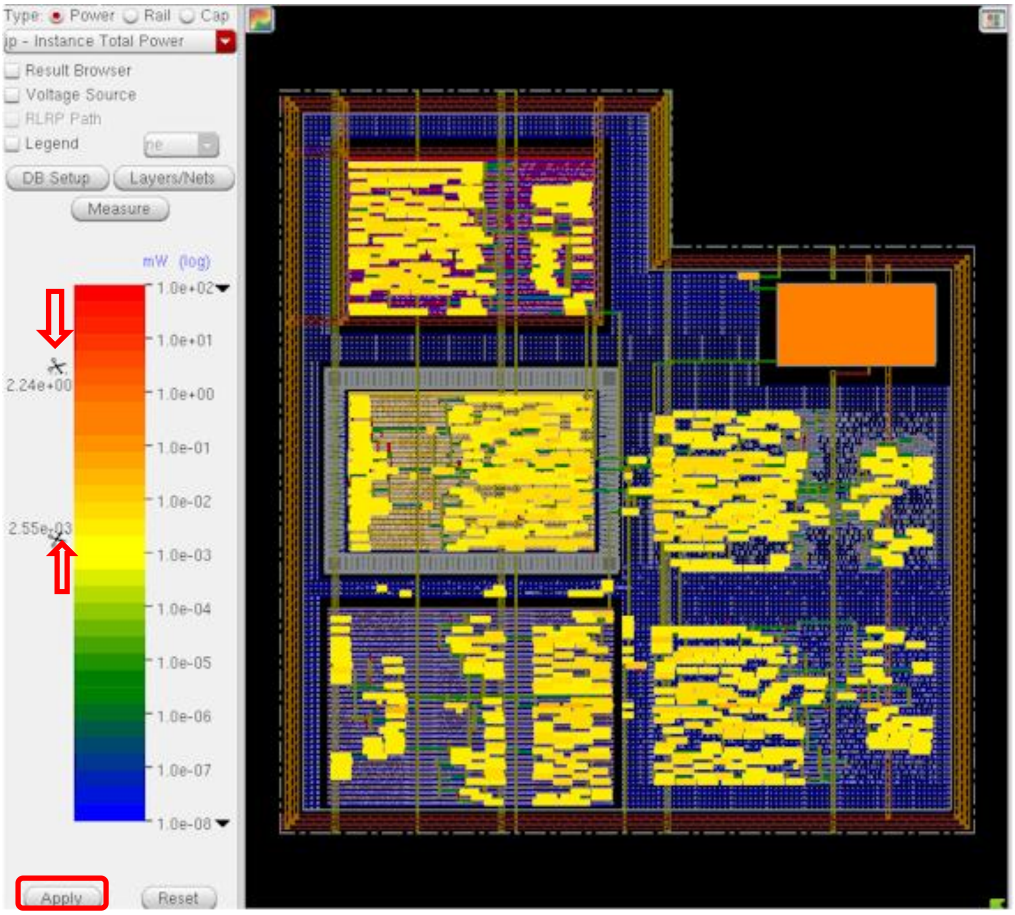
☑ Result Browser

□ Voltage Source

□ RLRP Path

□ Legend

9. Adjusting filter: on the left side of the Voltus debug panel, you can adjust the filter range by moving the top and bottom cursors up and down, then click *Apply* button:



Type: ● Power ○ Rail ○ Cap

ip - Instance Total Power

□ Result Browser

□ Voltage Source

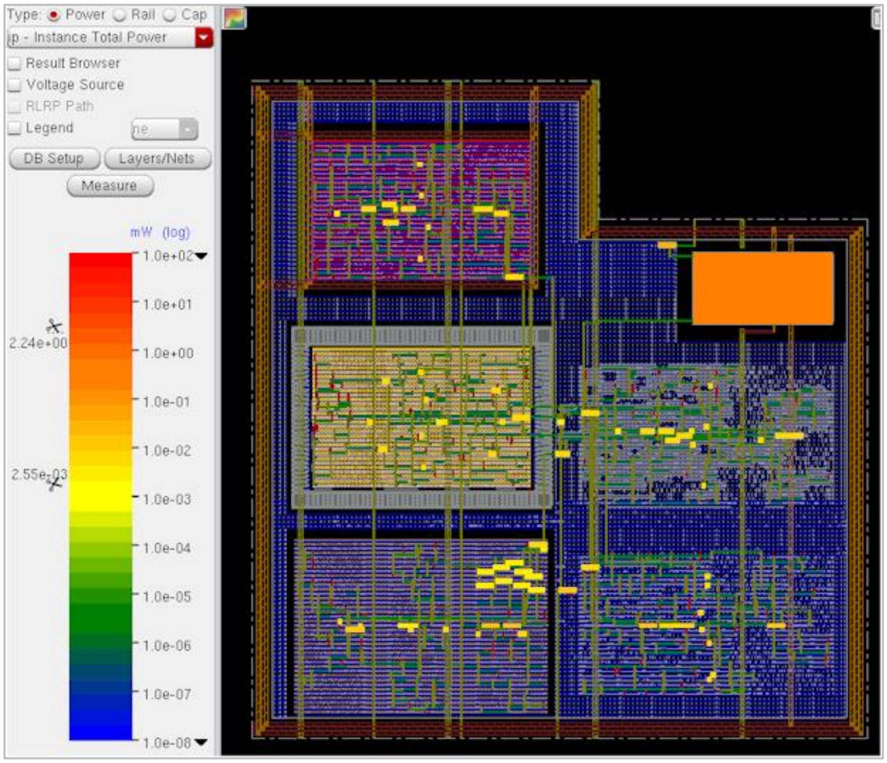
□ RLRP Path

□ Legend

TCL command:

```
report_power_rail_results \
   -plot                ip  \
   -filter_max          2.24  \
   -filter_min          2.55e-03
```

After applying the filter, the plot will appear as in the following image:

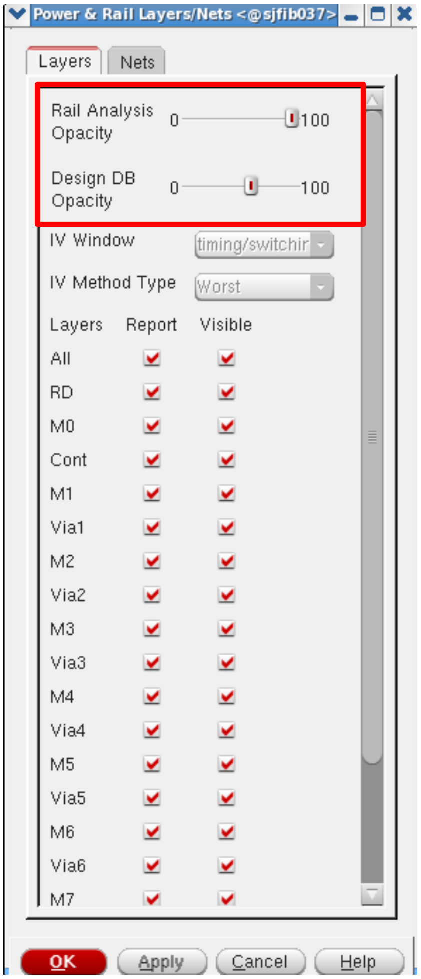


Type: ● Power ○ Rail ○ Cap

ip - Instance Total Power

10. “*Layer and Nets*” function:

    If you find it difficult to see the results, click on the “Layer & Nets” button to bring the following form up.



Rail Analysis Opacity 0 100

Design DB Opacity 0 100

IV Window timing/switchir

IV Method Type Worst

Layers	Report	Visible
All	✓	✓
RD	✓	✓
M0	✓	✓
Cont	✓	✓
M1	✓	✓
Via1	✓	✓
M2	✓	✓
Via2	✓	✓
M3	✓	✓
Via3	✓	✓
M4	✓	✓
Via4	✓	✓
M5	✓	✓
Via5	✓	✓
M6	✓	✓
Via6	✓	✓
M7	✓	✓

a) Move the "*Rail Analysis Opacity*" slider to adjust the transparency.

b) Move the "*Design DB Opacity*" slider to the left to hide the design and focus on the results.

11. The table at the beginning of this section provides other plot types. Experiment with different plots such as the instance dominant frequency and transition density:

```
report_power_rail_results \
    -plot       freq

report_power_rail_results \
    -plot       td
```

12. You can clear the display with:

```
report_power_rail_results \
    -plot       none
```

Static Rail Analysis

Setup and run static rail analysis

Now that you have completed power analysis, use the current each instance consumes to drive rail analysis (and to compute IR drop).

1. First, set the rail analysis mode:

```
set_rail_analysis_mode \
    -method                  static \
    -accuracy                hd \
    -analysis_view           AV_wc_on \
    -power_grid_library      { \
                             ../data/pgv_dir/tech_pgv/techonly.cl \
                             ../data/pgv_dir/stdcell_pgv/stdcells.cl \
                             ../data/pgv_dir/macro_pgv/macros_pll.cl \
                             } \
    -use_em_view_list        ../data/voltus/em_view.list \
    -enable_rlrp_analysis    true \
    -verbosity               true \
    -temperature             125
```

What do these options mean?

`method` This specifies whether you are doing static or dynamic rail analysis. In this case, it is static analysis.

`accuracy`

Xd uses early view for all cells (fast performance)

hd uses IR view for macros and EM view for EM check

`analysis_view` This is used by Voltus to determine which power domains are on or off.

`power_grid_library` This is the path to the libraries to be used during analysis. Note that when you characterized the libraries, the directories had a .cl extension.

`use_em_view_list` Define which view types to use:

-use_early_view_list filename

-use_ir_view_list filename

-use_em_view_list filename

	Example: For the PLL, we use the EM view.
`enable_rlrp_analysis`	Enable resistivity analysis, default is false

2. Optional: If there is no CPF loaded, use the *`set_pg_nets`* command to define the power nets in the design, and assign some attributes to these nets. Then, use *`set_rail_analysis_domain`* to define power domains:

```
set_pg_nets -net VDD_external -voltage 0.9 -threshold 0.81
set_pg_nets -net VSS          -voltage 0.0 -threshold 0.09
set_pg_nets -net VDD_ring     -voltage 0.9 -threshold 0.81
set_pg_nets -net VDD_AO       -voltage 0.9 -threshold 0.81
set_pg_nets -net VDD_column   -voltage 0.9 -threshold 0.81

set_rail_analysis_domain -name PD_external      \
      -pwrnets VDD_external  -gndnets VSS -threshold 0.10
set_rail_analysis_domain -name PD_ring          \
      -pwrnets VDD_AO  -gndnets VSS -threshold 0.10
set_rail_analysis_domain -name PD_column        \
      -pwrnets VDD_AO  -gndnets VSS -threshold 0.10
set_rail_analysis_domain -name PD_AO            \
      -pwrnets VDD_AO  -gndnets VSS -threshold 0.10
set_rail_analysis_domain -name ALL              \
      -pwrnets {VDD_external VDD_AO} -gndnets VSS -threshold 0.10
```

`voltage`	This is the nominal voltage used during the analysis.
`threshold`	This is the minimum allowed voltage on the power net (or max rise on ground net). It is used to set filter thresholds for IR plots.

3. To perform rail analysis, you must define where the voltage sources each power/ground net are located. This is done with a power pad location file.

   When there are four domains, why do we specify only three? One is for VSS, which is common to all domains. And although there are four power domains, two (*PD_ring* and *PD_column*) are connected to the PD_AO domain through power shut off switches. It will be useful to consider the drop through the switches, as if it is one large domain. That is the reason why these three domains only require one power pad location file. The domain *PD_external* is connected directly to an outside source. Hence, it needs a power pad location file. Here is specifying three power pad location files:

```
set_power_pads -reset
set_power_pads \
    -net     VDD_AO \
    -format  xy \
    –file    ../design/super_filter_VDD_AO.pp

set_power_pads \
    -net     VDD_external \
   -format   xy \
   -file     ../design/super_filter_VDD_external.pp

set_power_pads \
    -net     VSS \
    -format  xy \
    -file    ../design/super_filter_VSS.pp
```

The format of the file is:

```
*vsrc_name        x           y           layer_name
vsrc3             22.215      374.800     Metal5
```

In this lab, only one voltage source per power net will be used. This will help you highlight an IR drop gradient across the chip. However, in a real design, hundreds of voltage sources may be required.

4. Next, you need to point the binary current files created during power analysis:

```
set_power_data –reset
set_power_data \
    -format current { \
        staticPowerResults/static_VDD_AO.ptiavg \
        staticPowerResults/static_VDD_column.ptiavg \
        staticPowerResults/static_VDD_ring.ptiavg \
        staticPowerResults/static_VDD_external.ptiavg \
        staticPowerResults/static_VSS.ptiavg \
    }
```

5. Finally, you can issue the `analyze_rail` command. There are many ways to call `analyze_rail`. You can specify a specific power net or domain. In this case, there are multiple domains that must be considered at once because the domains are partitioned by the power switches. To analyze all domains at once, issue the following:

```
analyze_rail -results_directory ./staticRailResults  -type domain ALL
```

6. To analyze only one domain, execute the following:

```
analyze_rail -results_directory ./staticRailResults -type domain PD_AO
```

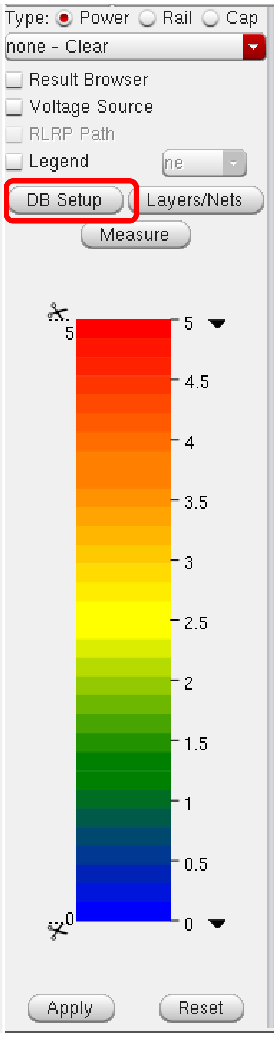
7. Alternatively, if the design did not have CPF and power domains, you can specify –type net (instead of domain) and the power net name:

```
analyze_rail -results_directory ./staticRailResults -type net VDD_AO
```

View rail analysis results

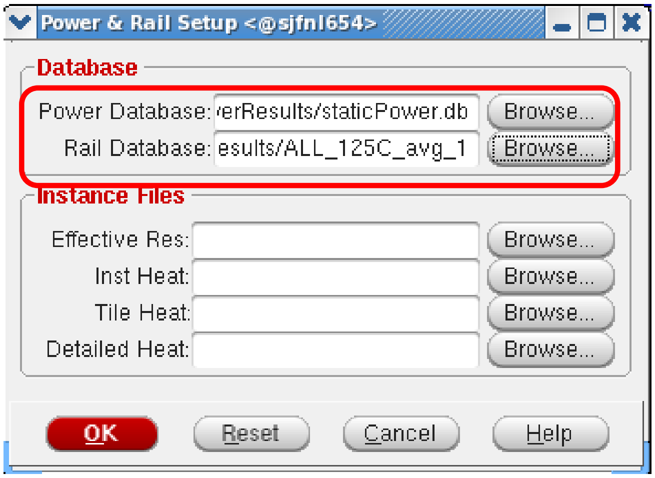
1. Load rail analysis results into Voltus through the Voltus debug panel:

    a. Go to *Power & Rail -> Power & Rail Plots* to bring it up on the left side of the design.



    b. Click the DB *Setup* button to bring up "*Power and Rail Setup*" form, and fill in both power and rail analysis data: *staticPowerResults/staticPower.db* and *staticRailResults/ALL_125C_avg_1*.

    c. Click *Ok*.



Power Database: verResults/staticPower.db

Rail Database: esults/ALL_125C_avg_1

Effective Res:

Inst Heat:

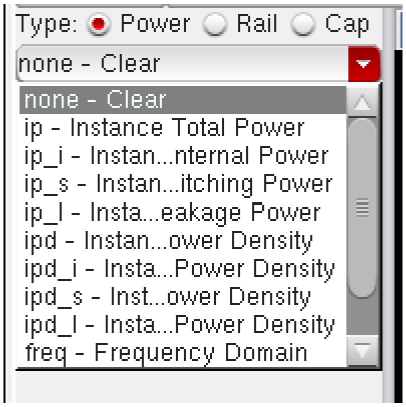
Tile Heat:

Detailed Heat:

TCL command to load power and rail analysis command:

```
read_power_rail_results
```

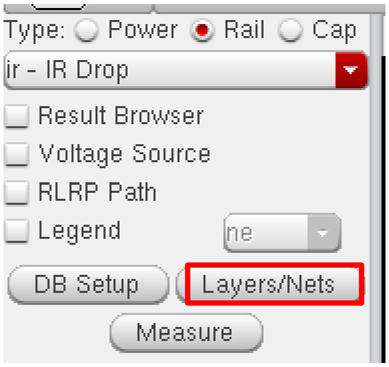
2. Display power results: This was covered in static power analysis section.


none - Clear
ip - Instance Total Power
ip_i - Instan...nternal Power
ip_s - Instan...itching Power
ip_l - Insta...eakage Power
ipd - Instan...ower Density
ipd_i - Insta...Power Density
ipd_s - Inst...ower Density
ipd_l - Insta...Power Density
freq - Frequency Domain

Type: ● Power ○ Rail ○ Cap

none - Clear

3. Layers & Net selection: click the "*Layers/Nets*" button. This will bring up *Power & Rail Layers/ Nets* form:



Type: ○ Power ● Rail ○ Cap

ir - IR Drop

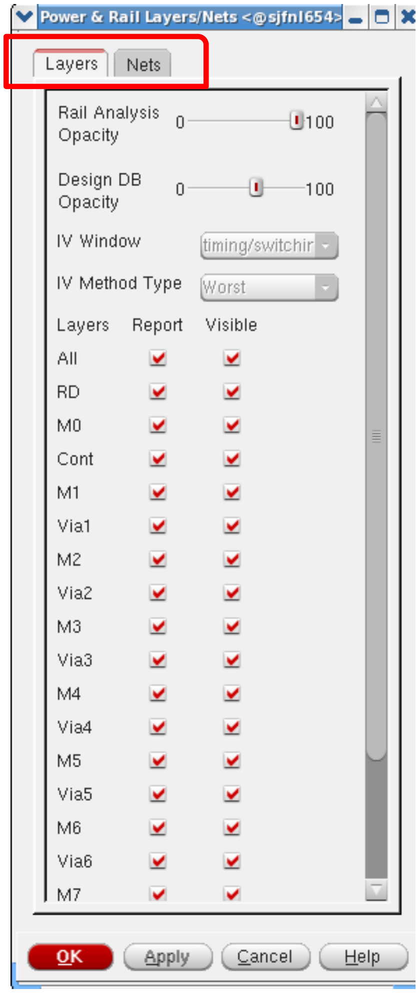
□ Result Browser

□ Voltage Source

□ RLRP Path

□ Legend ne

The *Layers* tab will let you to adjust layer selection:



Rail Analysis Opacity 0 100

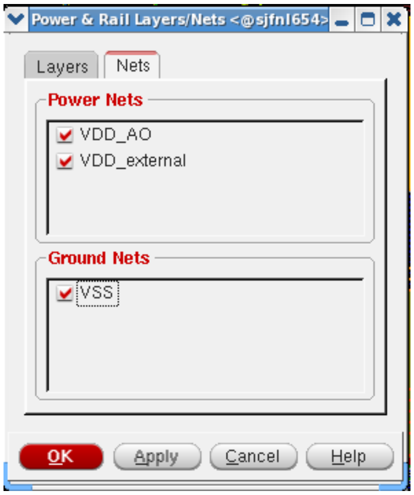
Design DB Opacity 0 100

IV Window: timing/switchir

IV Method Type: Worst

Layers	Report	Visible
All	☑	☑
RD	☑	☑
M0	☑	☑
Cont	☑	☑
M1	☑	☑
Via1	☑	☑
M2	☑	☑
Via2	☑	☑
M3	☑	☑
Via3	☑	☑
M4	☑	☑
Via4	☑	☑
M5	☑	☑
Via5	☑	☑
M6	☑	☑
Via6	☑	☑
M7	☑	☑

Click *Nets* tab to open the following form, which will let you select nets you want to see:



Power Nets

☑ VDD_AO

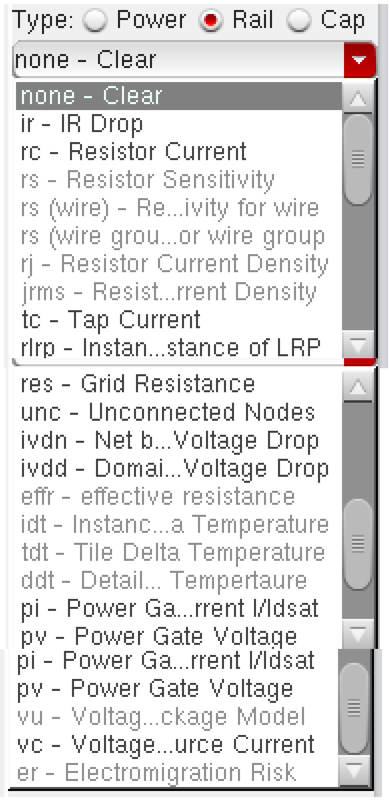
☑ VDD_external

Ground Nets

☑ VSS

The equivalent TCL command is `set_power_rail_layers_nets.`

4. Display Rail analysis results:


Type: ○ Power ● Rail ○ Cap
none - Clear
none - Clear
ir - IR Drop
rc - Resistor Current
rs - Resistor Sensitivity
rs (wire) - Re...ivity for wire
rs (wire grou...or wire group
rj - Resistor Current Density
jrms - Resist...rrent Density
tc - Tap Current
rlrp - Instan...stance of LRP
res - Grid Resistance
unc - Unconnected Nodes
ivdn - Net b...Voltage Drop
ivdd - Domai...Voltage Drop
effr - effective resistance
idt - Instanc...a Temperature
tdt - Tile Delta Temperature
ddt - Detail... Tempertaure
pi - Power Ga...rrent I/Idsat
pv - Power Gate Voltage
pi - Power Ga...rrent I/Idsat
pv - Power Gate Voltage
vu - Voltag...ckage Model
vc - Voltage...urce Current
er - Electromigration Risk

a. In the "*Type*" section of the window, click "*Rail*". By default, `analyze_rail` command performs many analyses. However, focus on the follow few types:

Key word	**Description**
ir	IR drop
tc	Tap current distribution
rc	Resistor current
rlrp	Least resistive path
res	Grid resistance
unc	Unconnected nodes
Ivdn	Net-based instance voltage drop

Ivdd	Domain-based instance voltage drop
pi	Power gate current ratio I/Idsat
pv	Power gate voltage drop
Effr	Effective resistance

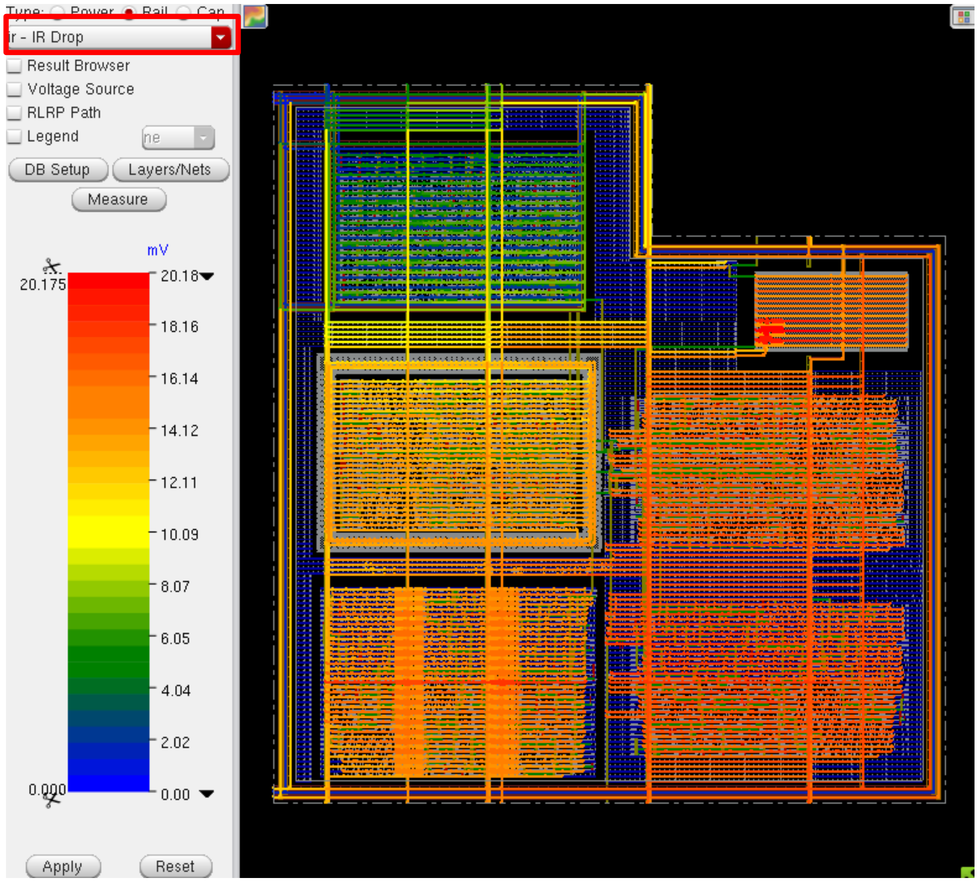
b. TCL command to display the plots:

```
report_power_rail_results -plot plot_type
```

5. First, select "*unc*" from the pull down. Any unconnected nodes will be highlighted with a red X. In this case, there are no unconnected nodes. However, this is an essential debug step. You must know whether any nodes are unconnected.

6. ir – IR Drop plot:

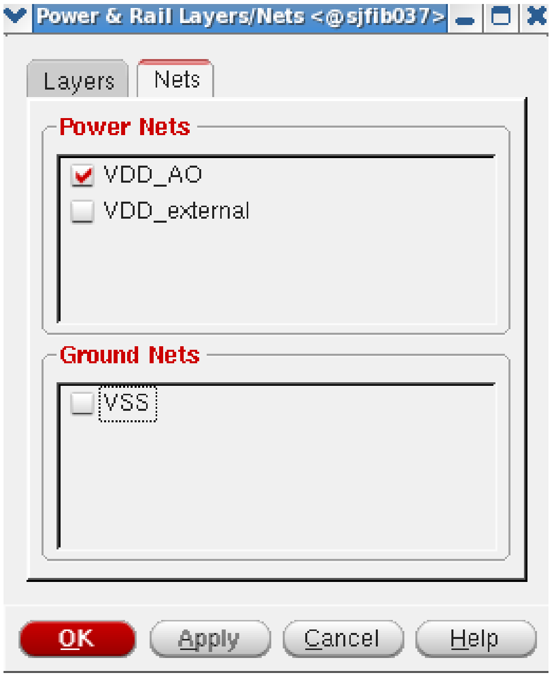
   a. Now, select "*ir – IR Drop*" from the pull down:



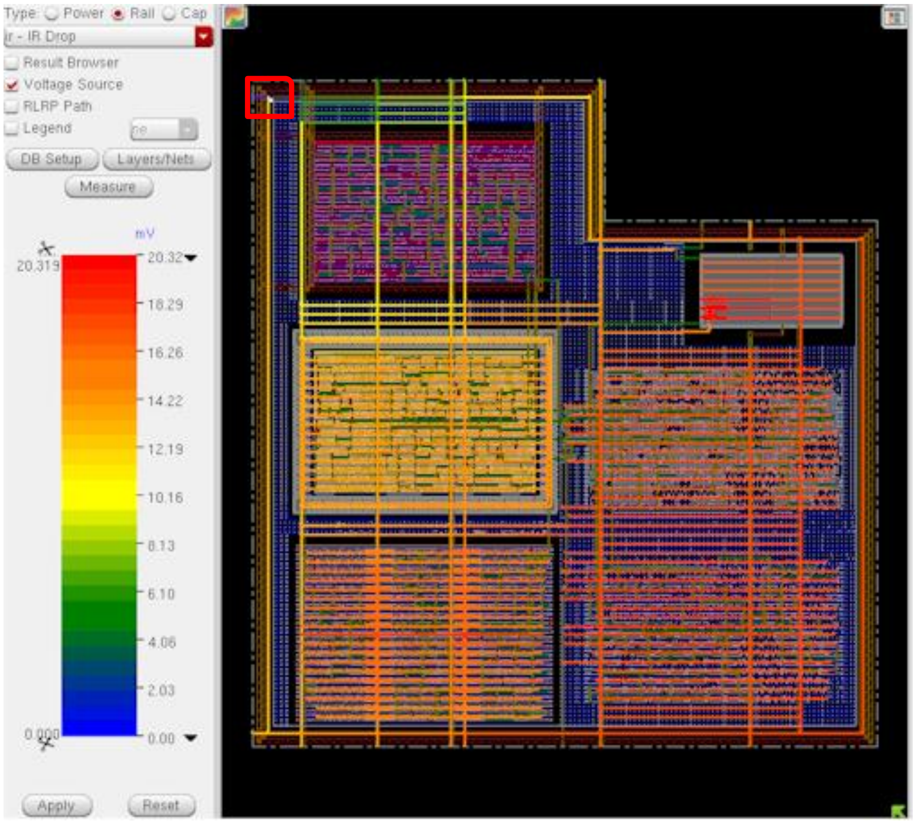
Voltus can display multiple nets by default. It displays all the nets for all the loaded data.

7. View net VDD_AO net with voltage source:

   a. Turn on voltage source display by clicking the *Voltage Source* button.
   b. Turn off net VSS and VDD_external on *Power & Rail Layers/Nets* form.



You will see the following VDD_AO ir drop plot where the voltage source is displayed as a white dot on the top left of the design:



Type: ○ Power ● Rail ○ Cap

ir - IR Drop

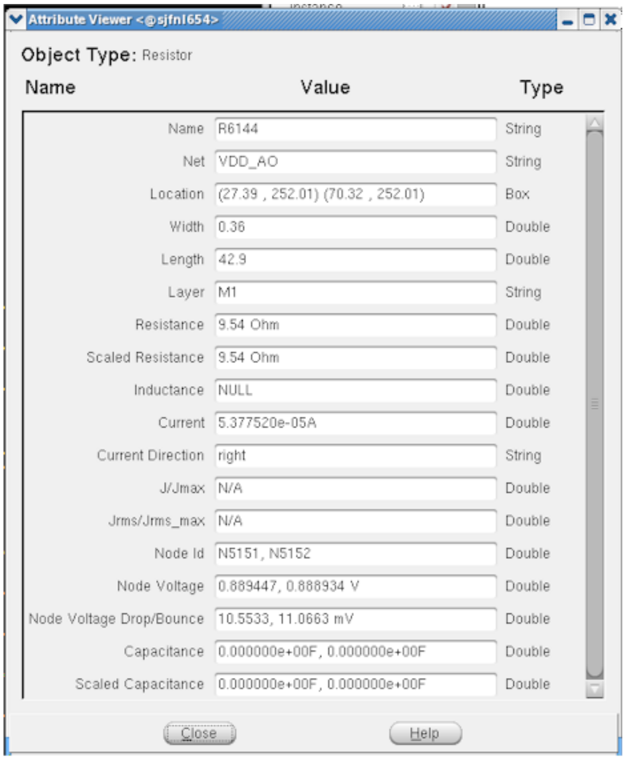
□ Result Browser

☑ Voltage Source

□ RLRP Path

□ Legend

8. View element property: Select an element in the design and press "*q*". You will get the following "*Attribute Viewer*", which provides a list of properties for the element selected:

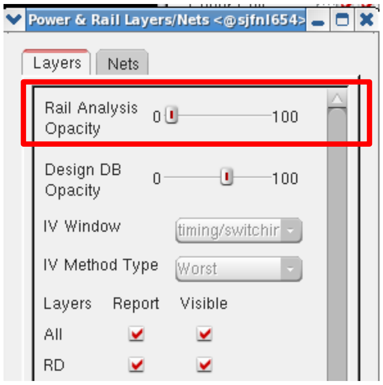


Attribute Viewer <@sjfnl654>

Object Type: Resistor

Name	Value	Type
Name	R6144	String
Net	VDD_AO	String
Location	(27.39 , 252.01) (70.32 , 252.01)	Box
Width	0.36	Double
Length	42.9	Double
Layer	M1	String
Resistance	9.54 Ohm	Double
Scaled Resistance	9.54 Ohm	Double
Inductance	NULL	Double
Current	5.377520e-05A	Double
Current Direction	right	String
J/Jmax	N/A	Double
Jrms/Jrms_max	N/A	Double
Node Id	N5151, N5152	Double
Node Voltage	0.889447, 0.888934 V	Double
Node Voltage Drop/Bounce	10.5533, 11.0663 mV	Double
Capacitance	0.000000e+00F, 0.000000e+00F	Double
Scaled Capacitance	0.000000e+00F, 0.000000e+00F	Double

9. tc – Tap Current plot:
    a. Chose plot type: *tc – Tap Current*.
    b. To see taps clearly, you might want to turn off visibility (For example, power domain, all layers in *Wire&Via* section)
    c. Adjust Rail Analysis Opacity in *Power & Rail Layers/Nets* display form as shown here:



Power & Rail Layers/Nets <@sjfnl654>

Rail Analysis Opacity 0 —— 100

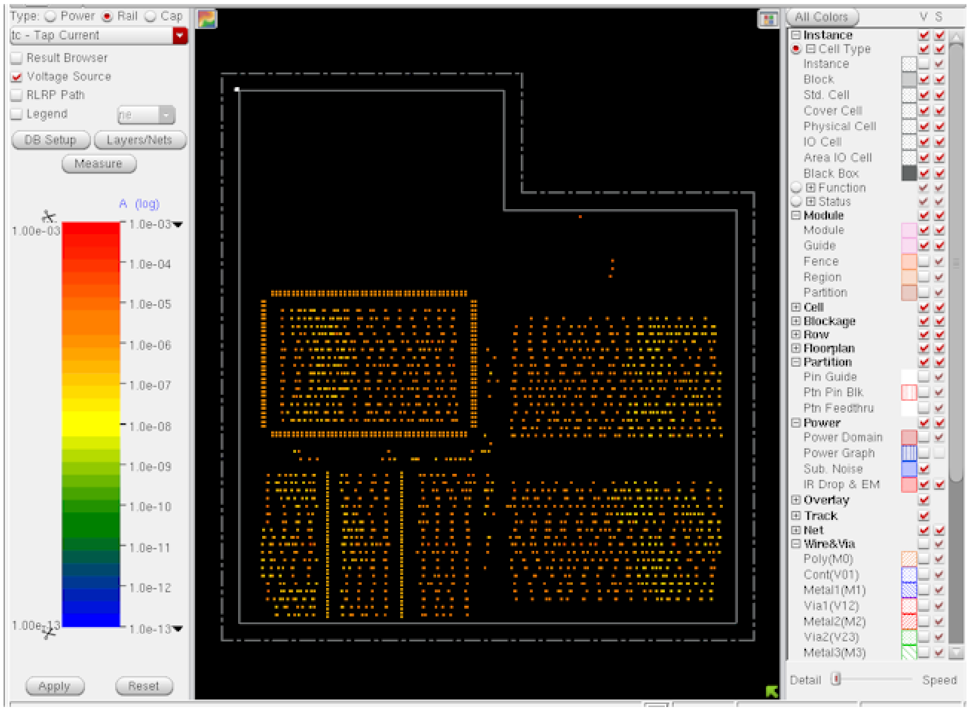
Design DB Opacity 0 —— 100

IV Window: timing/switchir

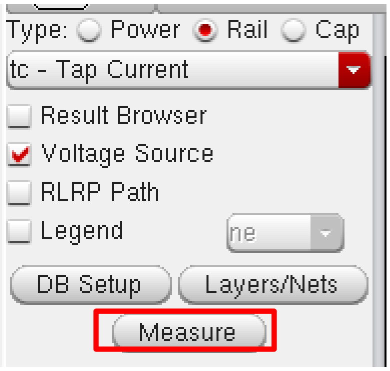
IV Method Type: Worst

Layers	Report	Visible
All	☑	☑
RD	☑	☑

    d. You will see the following *tc* plot, which shows the current consumption map for power net VDD_AO:



10. Measure function: There is a *Measure* button in debug pental.



a. Click the button and draw an area in the design. You will see the following info displayed in the terminal where you run Voltus:

```
Net: VDD_AO
Region: (20.842, 100.715) (47.0065, 115.105)
Tap Current:                1.16205e-05 A
Tap Current On Unconnected Grids:                0 A
Worst IR:                   0.884208 V
Grid Capacitance:           0 F
Intrinsic Capacitance(C1): 0 F
Load Capacitance(C2):       0 F
Decap Required:             0 F
```

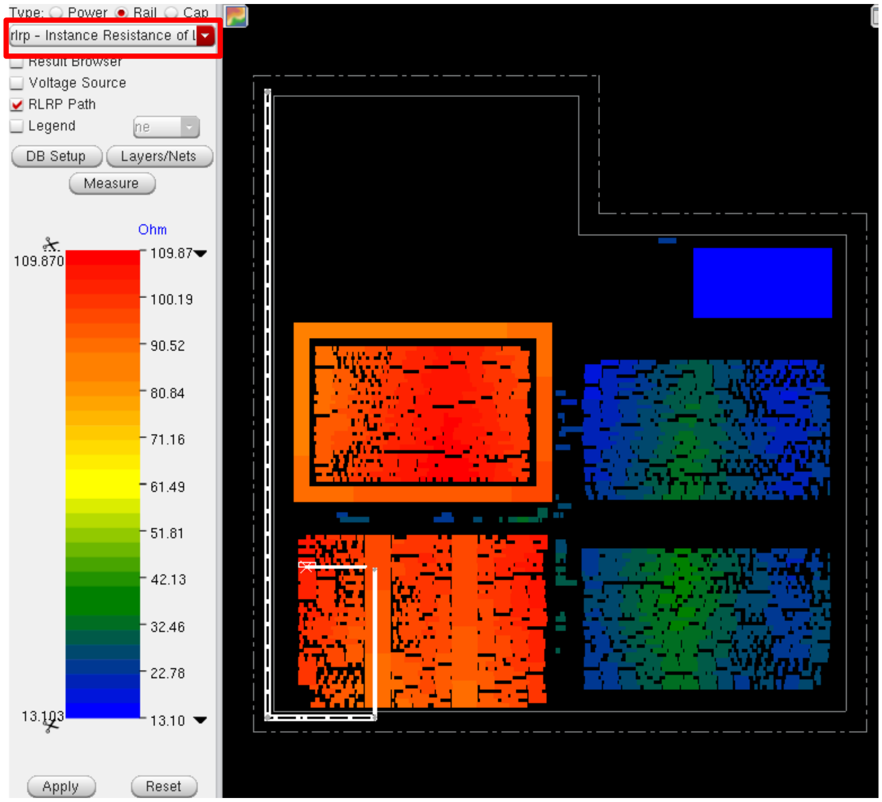
Because we are running static analysis, all the cap value is 0.

11. *Result Browser*: Try this feature to see what you can see and display.

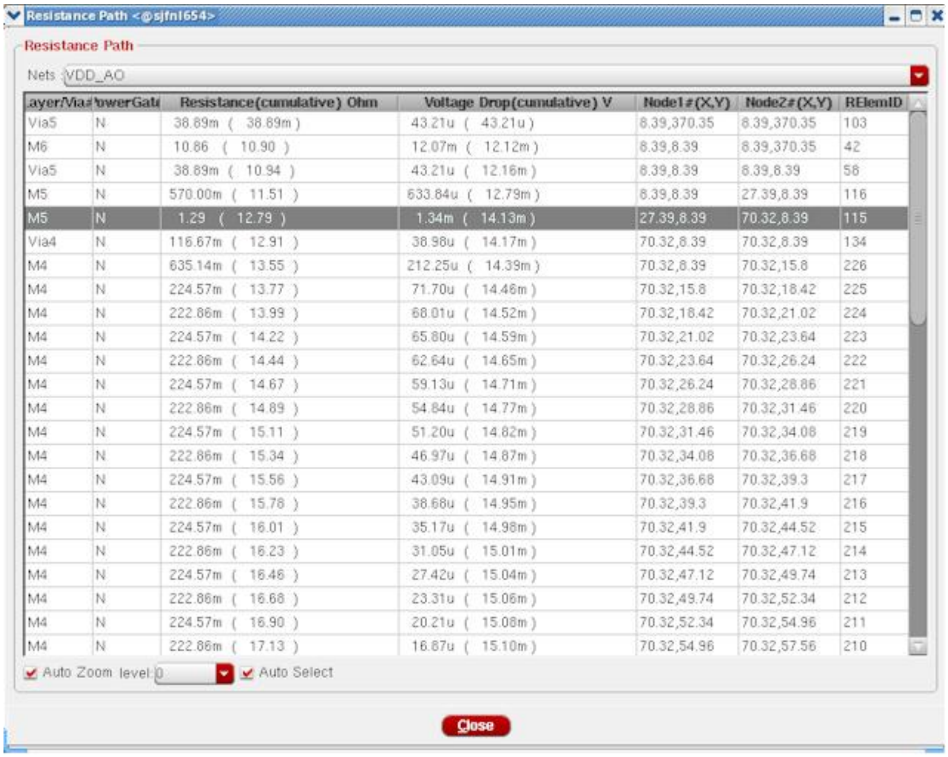
12. *rlrp* –instance Resistance of Least Resistive Path:

    a. Select *rlrp – Instance Resistance of LRP*.

    b. Select an instance in the design. You will see the following plot (the highlighted path is the least resistance path from the instance to the closest voltage source):

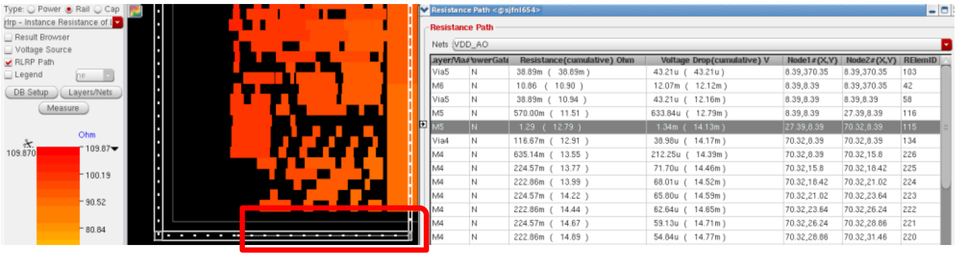


    c. The *Resistance Path* form will appear, which lists all the segments of the path:



.ayer/Via#	'owerGat	Resistance(cumulative) Ohm	Voltage Drop(cumulative) V	Node1#(X,Y)	Node2#(X,Y)	RElemID
Via5	N	38.89m (38.89m)	43.21u (43.21u)	8.39,370.35	8.39,370.35	103
M6	N	10.86 (10.90)	12.07m (12.12m)	8.39,8.39	8.39,370.35	42
Via5	N	38.89m (10.94)	43.21u (12.16m)	8.39,8.39	8.39,8.39	58
M5	N	570.00m (11.51)	633.84u (12.79m)	8.39,8.39	27.39,8.39	116
M5	N	1.29 (12.79)	1.34m (14.13m)	27.39,8.39	70.32,8.39	115
Via4	N	116.67m (12.91)	38.98u (14.17m)	70.32,8.39	70.32,8.39	134
M4	N	635.14m (13.55)	212.25u (14.39m)	70.32,8.39	70.32,15.8	226
M4	N	224.57m (13.77)	71.70u (14.46m)	70.32,15.8	70.32,18.42	225
M4	N	222.86m (13.99)	68.01u (14.52m)	70.32,18.42	70.32,21.02	224
M4	N	224.57m (14.22)	65.80u (14.59m)	70.32,21.02	70.32,23.64	223
M4	N	222.86m (14.44)	62.64u (14.65m)	70.32,23.64	70.32,26.24	222
M4	N	224.57m (14.67)	59.13u (14.71m)	70.32,26.24	70.32,28.86	221
M4	N	222.86m (14.89)	54.84u (14.77m)	70.32,28.86	70.32,31.46	220
M4	N	224.57m (15.11)	51.20u (14.82m)	70.32,31.46	70.32,34.08	219
M4	N	222.86m (15.34)	46.97u (14.87m)	70.32,34.08	70.32,36.68	218
M4	N	224.57m (15.56)	43.09u (14.91m)	70.32,36.68	70.32,39.3	217
M4	N	222.86m (15.78)	38.68u (14.95m)	70.32,39.3	70.32,41.9	216
M4	N	224.57m (16.01)	35.17u (14.98m)	70.32,41.9	70.32,44.52	215
M4	N	222.86m (16.23)	31.05u (15.01m)	70.32,44.52	70.32,47.12	214
M4	N	224.57m (16.46)	27.42u (15.04m)	70.32,47.12	70.32,49.74	213
M4	N	222.86m (16.68)	23.31u (15.06m)	70.32,49.74	70.32,52.34	212
M4	N	224.57m (16.90)	20.21u (15.08m)	70.32,52.34	70.32,54.96	211
M4	N	222.86m (17.13)	16.87u (15.10m)	70.32,54.96	70.32,57.56	210

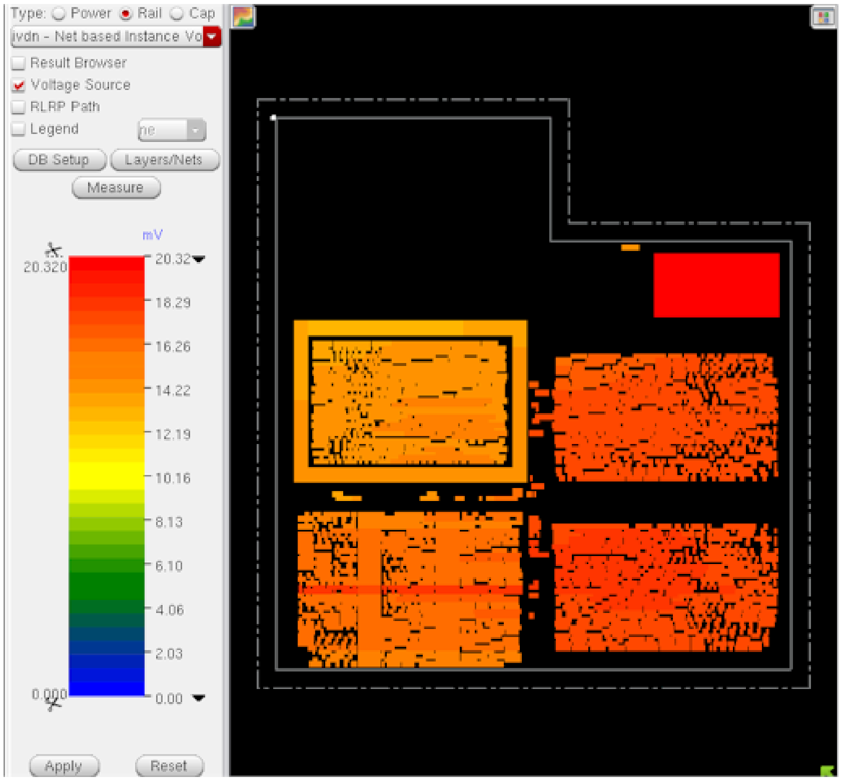
d. You can view each segment by selecting a segment of the path in the table, and Voltus will zoom in to the segment (Try a few):



.ayer/Via#	'owerGat	Resistance(cumulative) Ohm	Voltage Drop(cumulative) V	Node1#(X,Y)	Node2#(X,Y)	RElemID
Via5	N	38.89m (38.89m)	43.21u (43.21u)	8.39,370.35	8.39,370.35	103
M6	N	10.86 (10.90)	12.07m (12.12m)	8.39,8.39	8.39,370.35	42
Via5	N	38.89m (10.94)	43.21u (12.16m)	8.39,8.39	8.39,8.39	58
M5	N	570.00m (11.51)	633.84u (12.79m)	8.39,8.39	27.39,8.39	116
M5	N	1.29 (12.79)	1.34m (14.13m)	27.39,8.39	70.32,8.39	115
Via4	N	116.67m (12.91)	38.98u (14.17m)	70.32,8.39	70.32,8.39	134
M4	N	635.14m (13.55)	212.25u (14.39m)	70.32,8.39	70.32,15.8	226
M4	N	224.57m (13.77)	71.70u (14.46m)	70.32,15.8	70.32,18.42	225
M4	N	222.86m (13.99)	68.01u (14.52m)	70.32,18.42	70.32,21.02	224
M4	N	224.57m (14.22)	65.80u (14.59m)	70.32,21.02	70.32,23.64	223
M4	N	222.86m (14.44)	62.64u (14.65m)	70.32,23.64	70.32,26.24	222
M4	N	224.57m (14.67)	59.13u (14.71m)	70.32,26.24	70.32,28.86	221
M4	N	222.86m (14.89)	54.84u (14.77m)	70.32,28.86	70.32,31.46	220

13. *ivdn - Net base Instance Voltage Drop* plot:

a. Select *ivdn - Net base Instance Voltage* option from the pull down menu. You will see instance voltage plot for VDD_AO as shown in the following image (Feel free to see net based instance voltage drop for net VSS and VDD_external):



Type: ○ Power ● Rail ○ Cap

ivdn - Net based Instance Vo

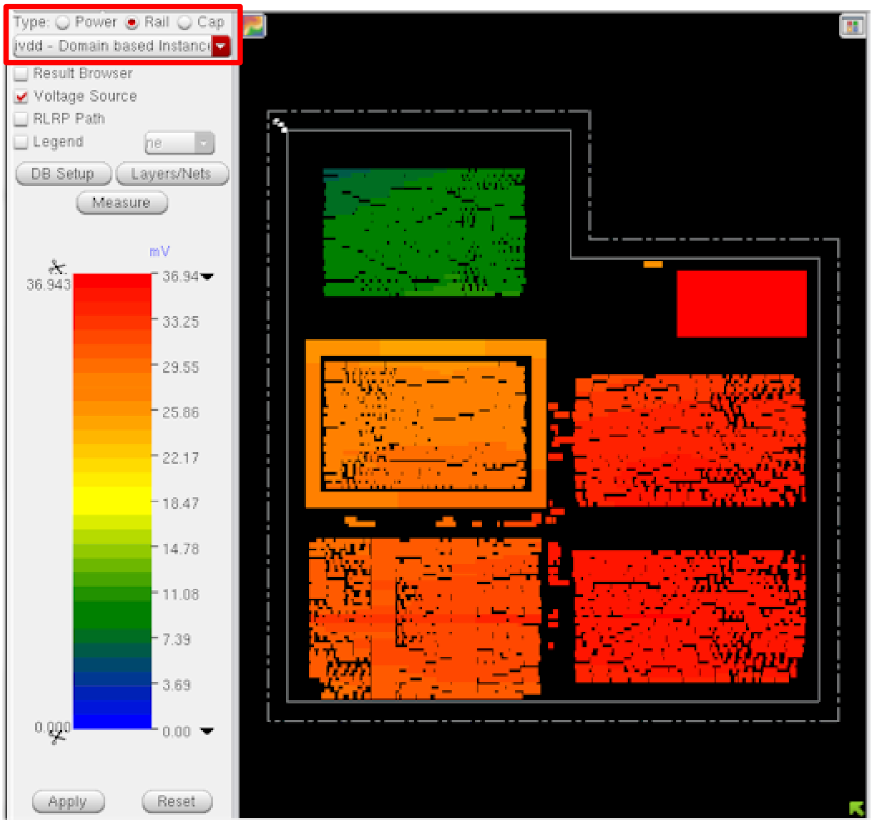
☐ Result Browser

☑ Voltage Source

☐ RLRP Path

☐ Legend

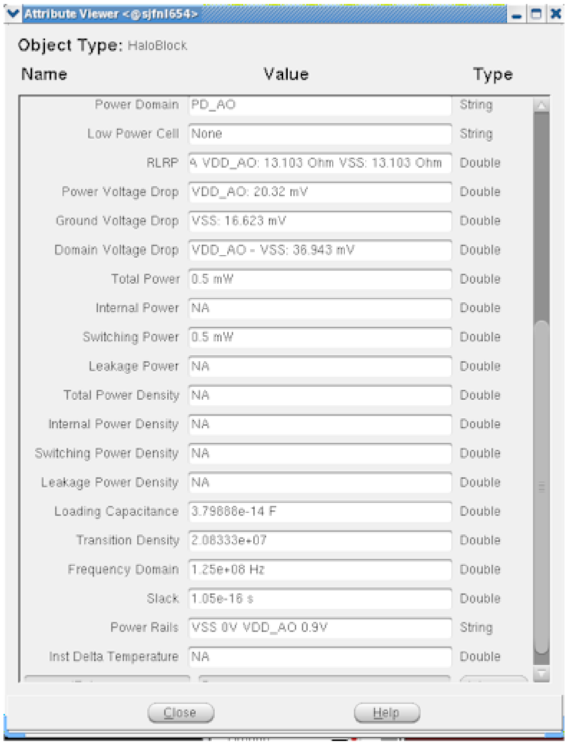
14. *ivdd – Domain based Voltage Drop* plot:
    a. To see domain-based instance voltage drop plot, you need to make sure all power and ground nets are turn on for display.
    b. Select the plot type:



Type: ○ Power ● Rail ○ Cap
vdd - Domain based Instance
☐ Result Browser
☑ Voltage Source
☐ RLRP Path
☐ Legend

c. View analysis result for an instance by selecting the instance and press "*q*".

d. Following is a list of analysis result for PLL:

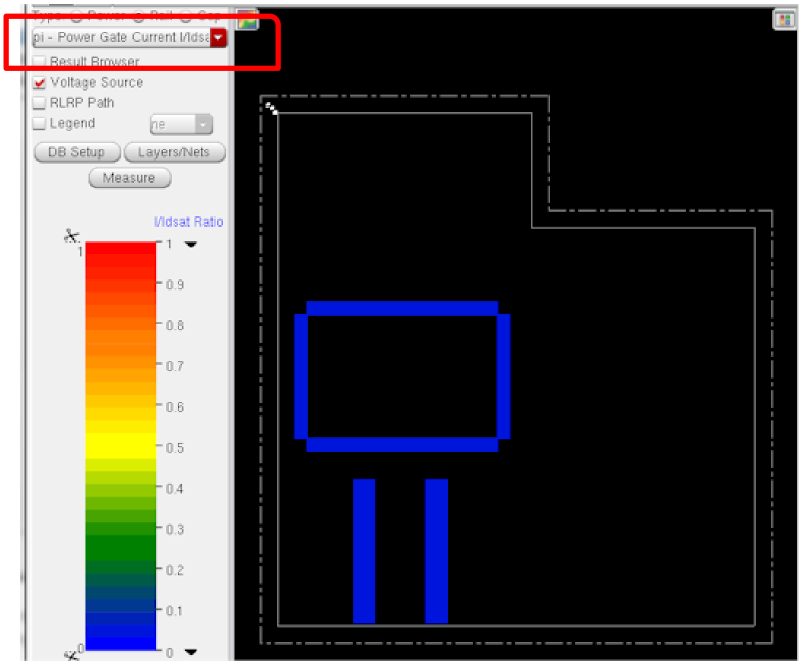


Object Type: HaloBlock

Name	Value	Type
Power Domain	PD_AO	String
Low Power Cell	None	String
RLRP	A VDD_AO: 13.103 Ohm VSS: 13.103 Ohm	Double
Power Voltage Drop	VDD_AO: 20.32 mV	Double
Ground Voltage Drop	VSS: 16.623 mV	Double
Domain Voltage Drop	VDD_AO - VSS: 36.943 mV	Double
Total Power	0.5 mW	Double
Internal Power	NA	Double
Switching Power	0.5 mW	Double
Leakage Power	NA	Double
Total Power Density	NA	Double
Internal Power Density	NA	Double
Switching Power Density	NA	Double
Leakage Power Density	NA	Double
Loading Capacitance	3.79888e-14 F	Double
Transition Density	2.08333e+07	Double
Frequency Domain	1.25e+08 Hz	Double
Slack	1.05e-16 s	Double
Power Rails	VSS 0V VDD_AO 0.9V	String
Inst Delta Temperature	NA	Double

15. pi – *Power Gate Current ratio I/Idsat* plot: I/Idsat current ratio for power gate cells:

    a. Select Rail plot type: *pi – Power Gate Current I/Idsat plot*.

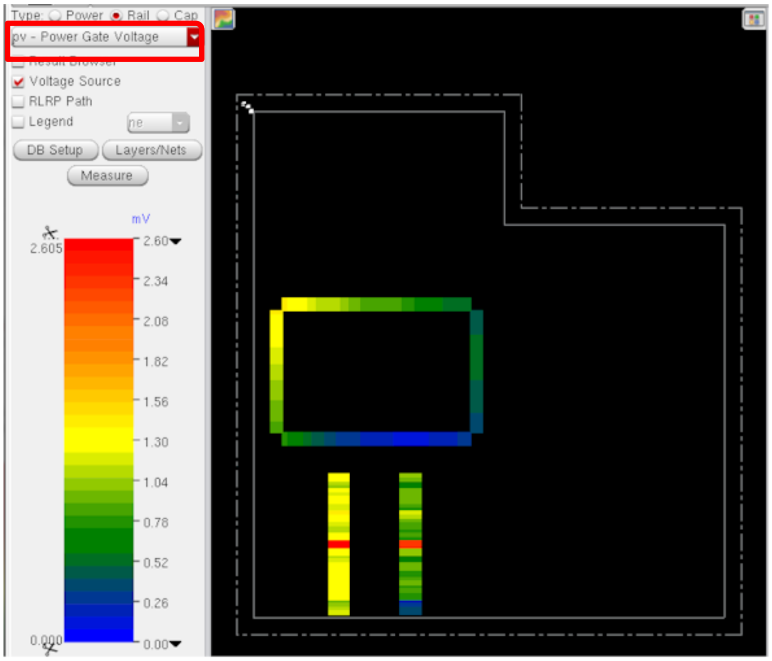


Type: ○ Power ○ Rail ○ Cap

pi - Power Gate Current I/Idsa

16. pv – *Power gate voltage drop* plot: voltage drop across power gate cells:

    a. Select Rail plot type: *pv – Power gate Voltage plot*:



Type: ○ Power ● Rail ○ Cap

pv - Power Gate Voltage

17. TCL command for generating plots:

    Again, it might be easier to create a script that creates the different plots. For example:

```
read_power_rail_results \
    -rail_directory             staticRailResults/ALL_125C_avg_1


report_power_rail_results \
      -plot                      ir
```

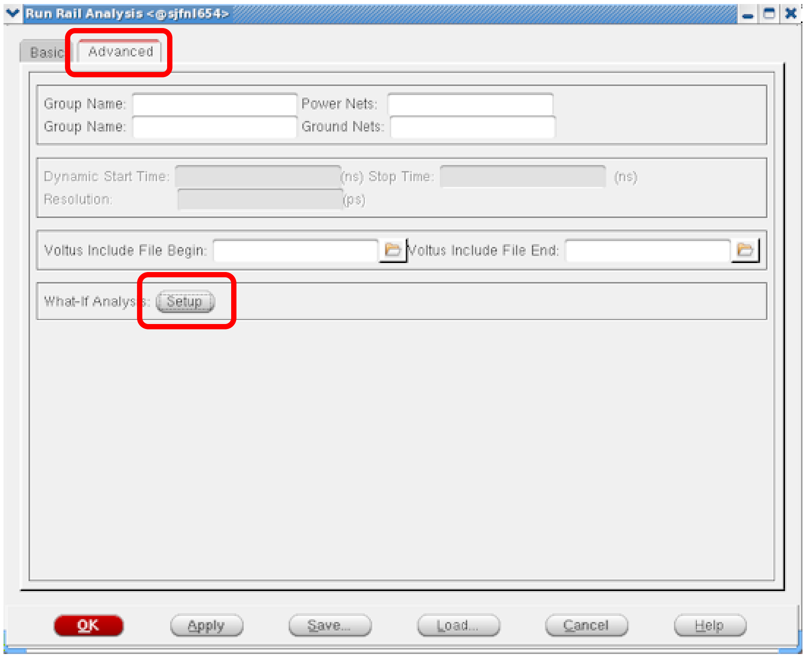
18. Early in the floorplanning process, it might be useful to see if it is possible to reduce the width of power rails (to reduce the area of the floorplan or to free up routing resources). You may have run rail analysis and discovered you need to increase the width of stripes but do not know by how much. What-if analysis makes this possible.

    What-if analysis helps you:

    a. Scale the resistance of the power grid
    b. Scale static and dynamic currents
    c. Scale capacitance to see the impact on dynamic IR drop analysis
    d. Create current regions to model unplaced instances or macros that are missing a power grid view

    Let us focus on scaling resistance and current, the first two What-if Analysis:

19. It is very easy to scale the resistance. The user has control of the metal layers, power net, and area to scale. To access the What-If GUI:
    a. Open the Run Rail Analysis window (Power & Rail->Run Rail Analysis).
    b. Click the “Advanced” tab.
    c. Click the “What-if Analysis” “Setup“.



d. A new window appears. Click the “Resistance” tab.

e. Click “Scale Resistance”.

f. Select Metal4 as the layer.

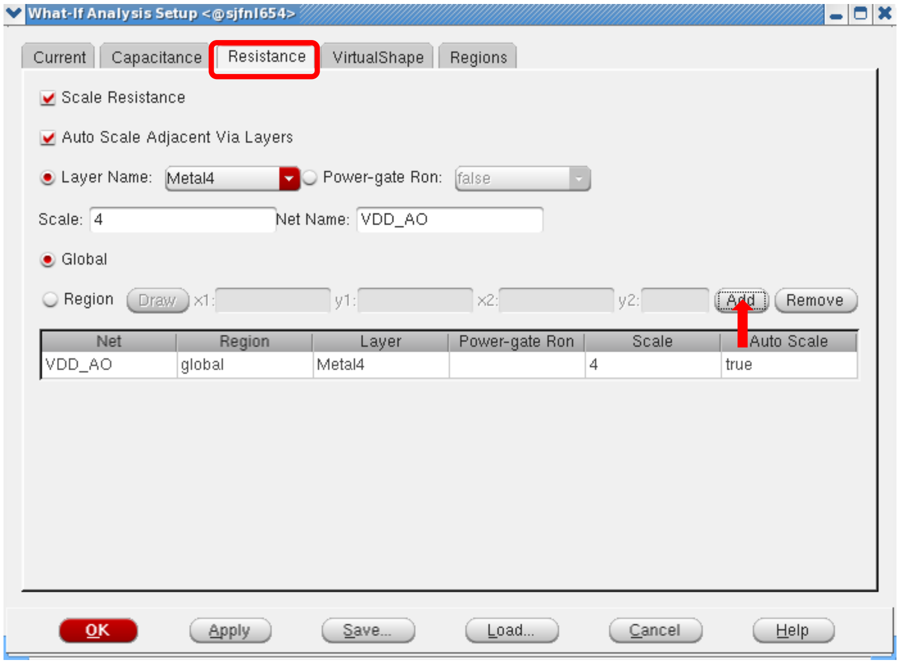
g. Set the scale value to 40 (to make resistance 40x greater, the same as reducing the width to 1/40 the existing width).

h. Select the net VDD_AO.

i. Click “Global” to scale all the Metal4 rings/stripes/etc.

j. Click the “Add” button to commit the changes.

k. The window should look like:



What-If Analysis Setup <@sjfnl654>

☑ Scale Resistance

☑ Auto Scale Adjacent Via Layers

● Layer Name: Metal4 ○ Power-gate Ron: false

Scale: 4 Net Name: VDD_AO

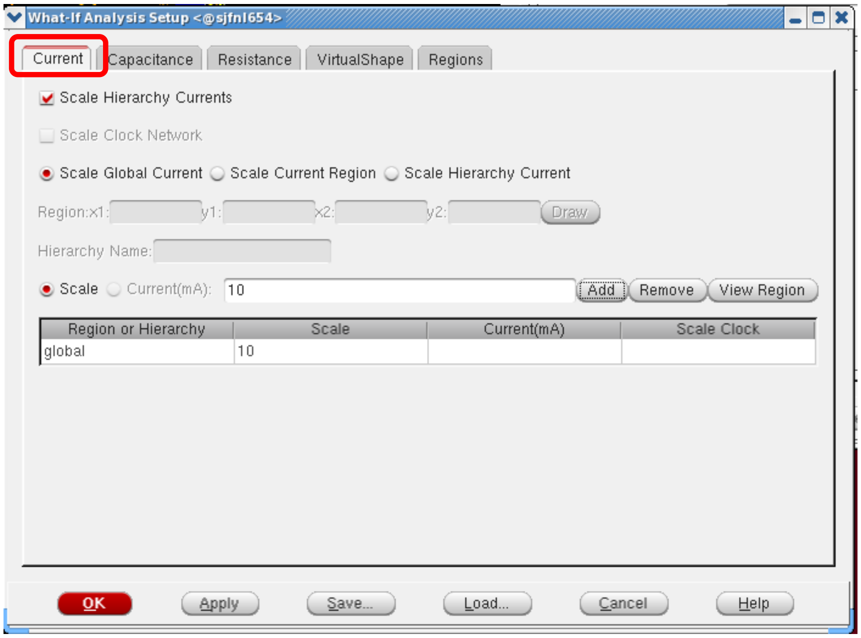
● Global

○ Region Draw x1: y1: x2: y2: Add Remove

Net	Region	Layer	Power-gate Ron	Scale	Auto Scale
VDD_AO	global	Metal4		4	true

20. Now, scale the current of the design:

    a. Click the “current” tab.
    b. Turn on “Scale Hierarchy Currents”.
    c. Keep “Scale Global Current” enabled (so all currents will be affected).
    d. Set the scale to 10 (to multiply the current by 10x).
    e. Click the “Add” button to commit the changes.
    f. The window should look like:



What-If Analysis Setup <@sjfnl654>

☑ Scale Hierarchy Currents

☐ Scale Clock Network

● Scale Global Current ○ Scale Current Region ○ Scale Hierarchy Current

Region:x1: y1: x2: y2: Draw

Hierarchy Name:

● Scale ○ Current(mA): 10 Add Remove View Region

Region or Hierarchy	Scale	Current(mA)	Scale Clock
global	10		

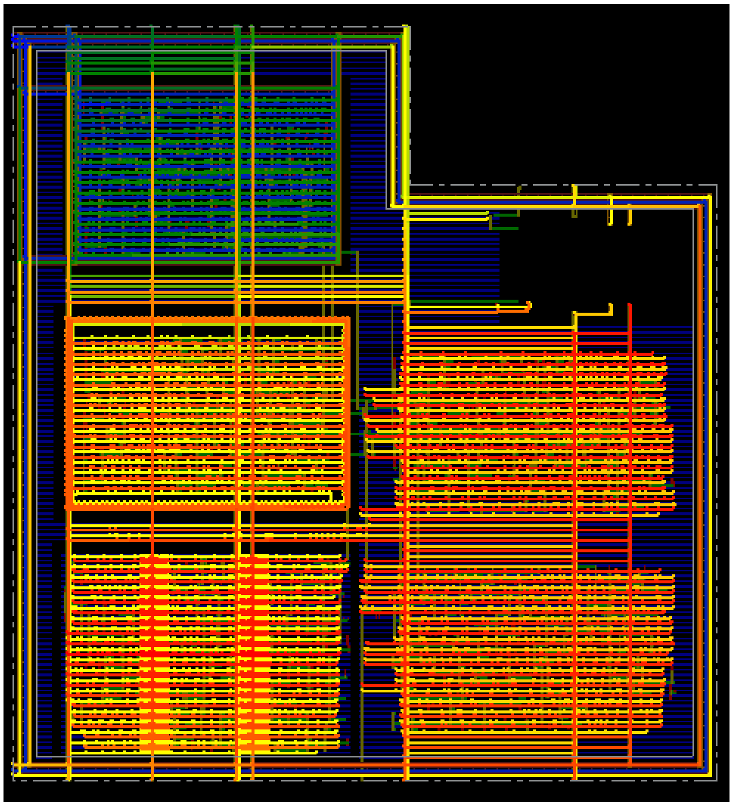
21. Click OK in the What-If window.
22. The equivalent TCL commands to scale the current and resistance:

```
scale_what_if_current \
    -global \
    -scale 10

scale_what_if_resistance \
    -global \
    -net                            VDD_AO \
    -layer                          Metal4 \
    -scale                          4 \
    -auto_scale_adjacent_via_layers  true
```

23. In the Voltus console, rerun the analysis by typing !analyze_rail.

24. When finished, invoke the Power & Rail Results browser.

    a. Open new rail analysis data using the DB Setup button.
    b. IMPORTANT: Change the state directory from the previous run to the latest. It should be staticRailResults/ALL_125C_avg_2.
    c. Select the “Rail Analysis Plot Type” to be the IR drop.
    d. In the layout view, you will see a worse than before IR drop plot.



e. Feel free to create current regions or change the resistance or current scale factors, and rerun rail analysis. When reviewing the results, remember to load in the correct rail analysis data base for plots.

25. Now, you can exit Voltus.

Dynamic Vectorless Power Analysis

Voltus supports two types of dynamic analysis: vector based and vectorless.

If you have VCD files, the vectors can drive a simulation to determine the current consumed by each instance over time. However, if you do not have vectors, Voltus can use the delays and transition times from static timing analysis to determine when the nets toggle, and use the toggle rates to statistically determine if a net will toggle. In the end, Voltus generates one dominant cycle's worth net transitions as if it were a simulation.

To prepare for dynamic vectorless power analysis:

1. Change directories:

```
cd dynamic
```

2. Start Voltus

```
voltus
```

3. Load the design like you did in the static analysis section:

```
source ../tcl/load_design.tcl
```

4. Set the sub directory that will contain the results:

```
set_power_output_dir dynVecLessPowerResults
```

5. Set the power analysis mode. This is essentially the same as in static power analysis. However, now, we change the method to dynamic_vectorless:

```
set_power_analysis_mode -reset
set_power_analysis_mode \
    -analysis_view          AV_wc_on \
    -disable_static         false \
    -write_static_currents  true \
    -binary_db_name         dynPower.db \
    -create_binary_db       true \
    -method                 dynamic_vectorless
```

analysis_view	Voltus can only analyze power for one view at a time. Here, we define the view to be analyzed.
disable_static	Default is true; need to set to false to output static data.

write_static_currents — This writes each instance of the current waveform to a binary database. For static power, this will be a single value (average). However, for dynamic power analysis, a current waveform will be saved.

You need to set “ `-disable_static false`” to generate this data.

create_binary_db — This specifies if the binary power db will be written. This db contains instances, locations and power. It allows for cross probing between the power debugger and the layout. You need to set “ `-disable_static false`” to generate this data.

binary_db_name — This is the name of the binary power database. This is not to be confused with the current file.

method — This specifies the type of analysis to be performed: static, dynamic_vectorless or dynamic.

6. In the previous section, you assigned an average power number to an instance. For dynamic analysis, we can assign a piece-wise linear waveform to an instance:

```
set_power –reset

# format
# set_power -pg_net <power pin> -instance <inst> {time current time current … …}

set_power -pg_net VDD_AO -pwl -instance ref_clk__L1_I0 \
    {0ns 0mA 0.075ns 0mA 0.175ns 45mA 0.225ns 15mA 0.425ns 0mA} -sticky
```

Here are some observations regarding piece-wise linear waveforms:

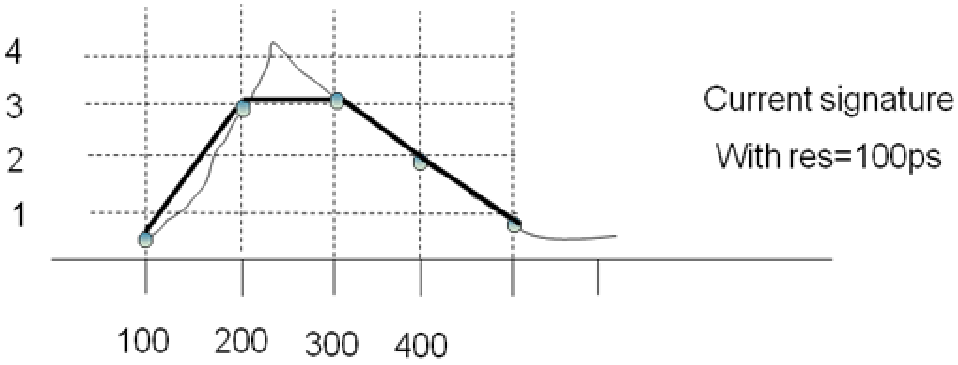
- Units are ALWAYS ns and mA.
- Omitting -sticky causes the waveform to be applied each time the output toggles.
- -sticky will apply the waveform as specified in the simulation. Note that if you are using a VCD file and start the simulation at some point other than t=0, the simulator will consider that start time t=0.

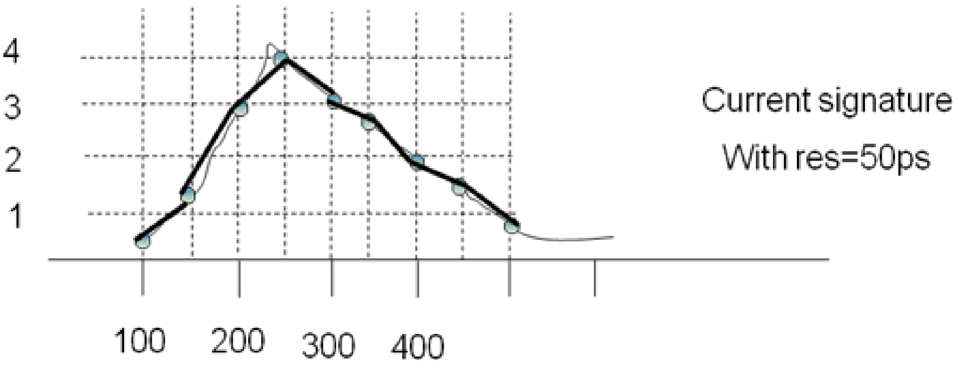
7. Define the simulation period and resolution:

```
set_dynamic_power_simulation -reset
set_dynamic_power_simulation -resolution 50ps
```

This is design-dependent, and may not need to be set. The default is 50ps. R&D recommends that it is set to a typical transition time. How can this affect the analysis?

You can see that if the resolution is not fine enough, current peaks may be truncated, affecting the accuracy of the analysis. The downside to having a small resolution is that the runtime will be negatively affected:





8. Now, run the power analysis:

```
report_power -outfile dyn.rpt
```

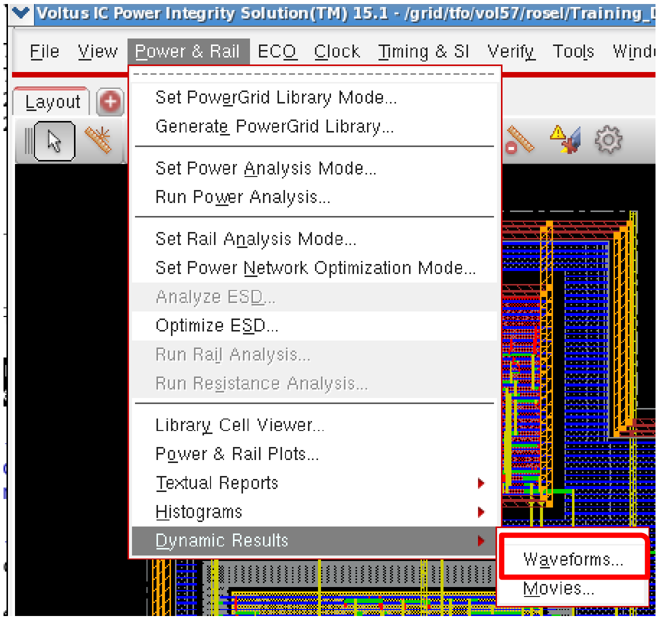
9. Earlier we set:

```
set_power_analysis_mode \
      -disable_static         false \
      -write_static_currents  true
```

Because of this, we will able look at the static power plots (like we did before).

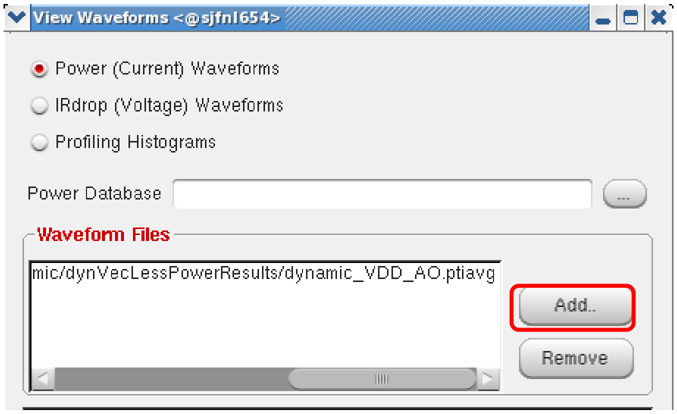
10. However, it is more interesting to look at current waveforms. Let us see what is the peak current for a particular instance, at the level of hierarchy or the whole design.

a. start_gui
b. Select Power & Rail -> Dynamic Results ->Waveforms.


Voltus IC Power Integrity Solution(TM) 15.1 - /grid/tfo/vol57/rosel/Training_
File
View
Power & Rail
ECO
Clock
Timing & SI
Verify
Tools
Windo
Layout
Set PowerGrid Library Mode...
Generate PowerGrid Library...
Set Power Analysis Mode...
Run Power Analysis...
Set Rail Analysis Mode...
Set Power Network Optimization Mode...
Analyze ESD...
Optimize ESD...
Run Rail Analysis...
Run Resistance Analysis...
Library Cell Viewer...
Power & Rail Plots...
Textual Reports
Histograms
Dynamic Results
Waveforms...
Movies...

c. Select "*Power (current) Waveforms*".
d. Under "*Waveform Files*", click add and select the file:

dynVecLessPowerResults/dynamic_VDD_AO.ptiavg



● Power (Current) Waveforms
○ IRdrop (Voltage) Waveforms
○ Profiling Histograms

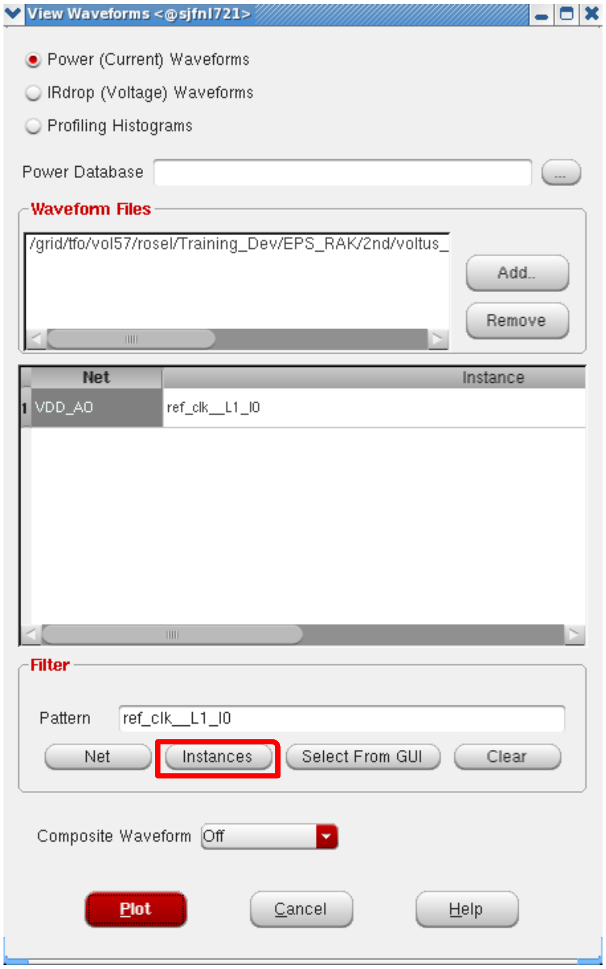
Power Database

Waveform Files

mic/dynVecLessPowerResults/dynamic_VDD_AO.ptiavg

Now you can query the waveforms of all the instances and nets of this power net.

e. In the filter section of the window, enter: *ref_clk__L1_I0*.
f. Click "*Instances*", and the instance *ref_clk__L1_I0* appears in the window.
g. Select this instance (so that it is highlighted). The window should look like this:


View Waveforms <@sjfnl721>

● Power (Current) Waveforms
○ IRdrop (Voltage) Waveforms
○ Profiling Histograms

Power Database

Waveform Files
/grid/tfo/vol57/rosel/Training_Dev/EPS_RAK/2nd/voltus_

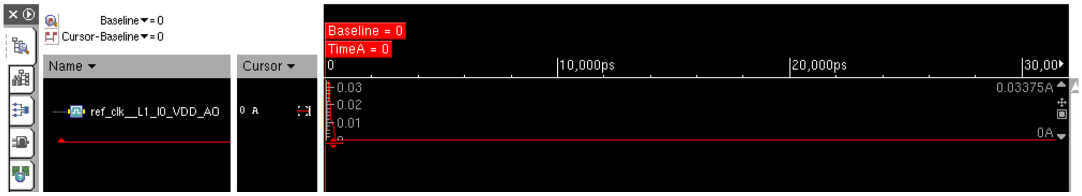
	Net	Instance
1	VDD_AO	ref_clk__L1_I0

Filter
Pattern ref_clk__L1_I0

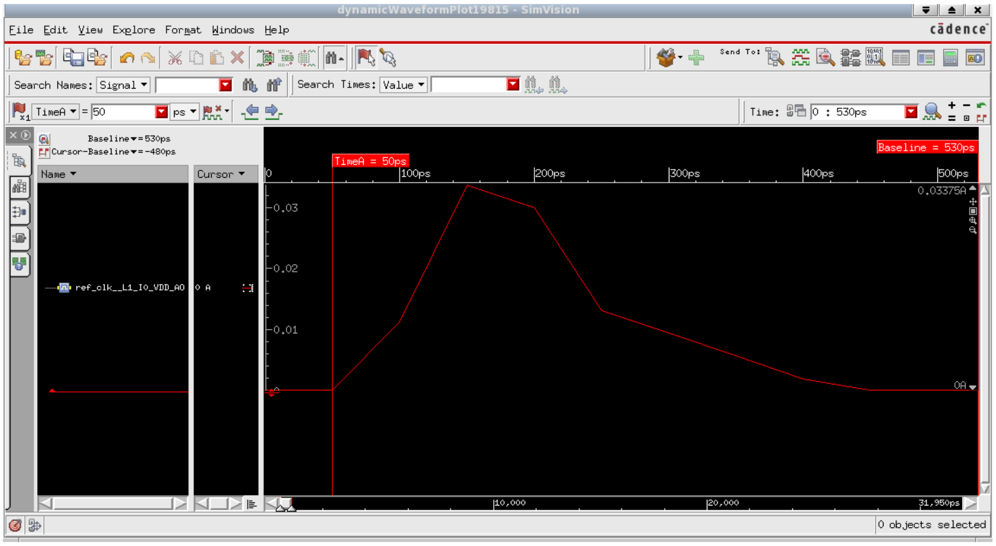
Composite Waveform Off

h. Click "*Plot*".

i. *SimVision* will open, but the waveform is not clearly visible:



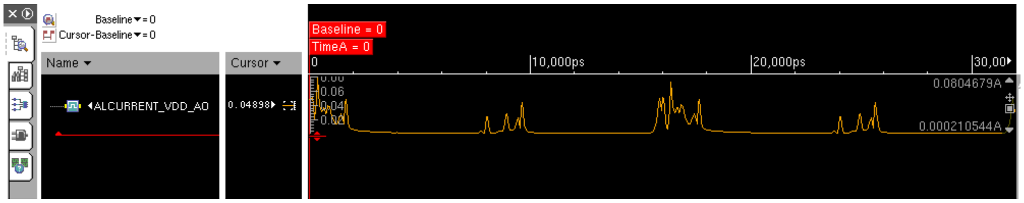
j. To improve visibility:
   i. *View -> Zoom -> Full Y*.
   ii. Place the cursor on the far left of the waveform to place a marker near 1000ps. Use the LMB to adjust the location of the "baseline".
   iii. Select *View -> Zoom -> Cursor-Baseline* (or control-alt).
   iv. Eventually, you should see a waveform like shown in the following image:



k. The command for displaying the current for this instance (so that you do not have to refer to the GUI) is:

```
view_dynamic_waveform \
  -type             current \
  -instance_name    ref_clk__L1_IO \
  -waveform_files   {dynVecLessPowerResults/dynamic_VDD_AO.ptiavg}
```

l. To display the current for the entire design:
   i. In the "*View Waveforms*" window, for the "*Composite Waveform*", select "*Total Current*".
   ii. Click "*Plot*". The waveform will appear in the existing *SimVision* window.
   iii. Select the old instance name in the plot and click the "delete" button to remove it from the display.
   iv. Select *View -> Zoom -> Full X*.
   v. Select *View -> Zoom -> Full Y*. The plot will look like this:



m. Why did the simulation run for 32ns? In the log file, you can see the various clocks that are detected by Voltus:

```
io_clk(125MHz)
   test_clk(62.5MHz)
   ref_clk(125MHz)
   pll_clk(62.5MHz)
** INFO:   (VOLTUS_POWR-2054): Setting default frequency to the
dominant frequency 62.5MHz.
```

You can see the dominant frequency is 62.5MHz, which is a 16ns period. By default, Voltus will simulate for 2x of the dominant clock frequency.

n. The command to display the current for this instance (so that you do not have to refer to the GUI) is:

```
view_dynamic_waveform \
   -type                      current \
   -composite_waveform_type  total_current \
   -waveform_files           {dynVecLessPowerResults/dynamic_VDD_AO.ptiavg}
```

o. Exit all SimVision windows.

p. Click “*cancel*” in the “*View Waveforms*” window.

11. Exit Voltus.

Dynamic Vector Based Power Analysis

When doing rail analysis, you get the most accurate results by providing a full VCD file to drive the simulation. Dynamic vectorless calculates the toggle rates and signal transition times based on the inputs that may lead to inaccuracies. In this section, you will read a VCD file to drive power analysis.

To prepare for dynamic vector power analysis:

If you did not exit Voltus, move to step 4. Otherwise, follow these steps to load design:

1. Go to dynamic directory:

   ```
   cd dynamic
   ```

2. Start Voltus:

   ```
   voltus
   ```

3. Load the design like we did in the dynamic vectorless analysis section:

   ```
   source ../tcl/load_design.tcl
   ```

4. Set the sub directory that will contain the results:

   ```
   set_power_output_dir dynPowerResults
   ```

5. Set the power analysis mode. This is essentially the same as in static power analysis. However, here, we change the method to dynamic_vectorless.

   ```
   set_power_analysis_mode -reset
   ```

6. Now set the power analysis mode as in the vectorless analysis. However, there is one difference. Change the method from dynamic_vectorless to dynamic_vectorbased:

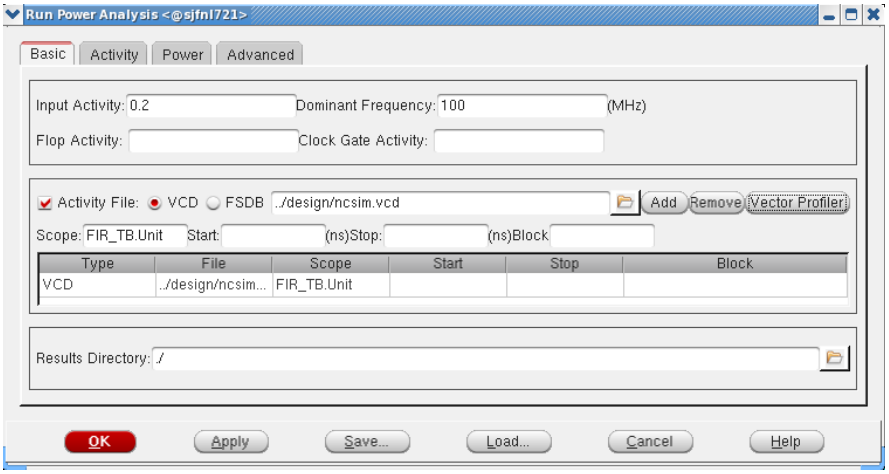
   ```
   set_power_analysis_mode \
       -analysis_view          AV_wc_on \
       -disable_static         false \
       -write_static_currents  true \
       -binary_db_name         dynPower.db \
       -create_binary_db       true \
       -method                 dynamic_vectorbased
   ```

7. Now set the resolution of the simulation:

```
set_dynamic_power_simulation -reset
set_dynamic_power_simulation -resolution 50ps
```

8. Now point to the VCD file you plan to use:

   a. Open *Power & Rail ->Run Power Analysis*.
   b. Click "*Activity File*".
   c. Select "*VCD*".
   d. Open the folder to select the VCD file. Point to the file *../design/ncsim.vcd*.
   e. Set the "*Scope*" to *FIR_TB.Unit*.
   f. Leave other fields blank.
   g. Click the "*Add*" button to add this VCD file to the list of stimulus.
   h. DO NOT Click *OK* or *Apply*.
   i. The resulting window should look like this:



Input Activity: 0.2 Dominant Frequency: 100 (MHz)

Flop Activity: Clock Gate Activity:

☑ Activity File: ● VCD ○ FSDB ../design/ncsim.vcd

Scope: FIR_TB.Unit Start: (ns)Stop: (ns)Block

Type	File	Scope	Start	Stop	Block
VCD	../design/ncsim...	FIR_TB.Unit			

Results Directory: ./

   j. Alternatively, you could issue the following command:

```
set_power_analysis_mode \
    -report_missing_nets  true

read_activity_file \
    -format                VCD \
    -scope                 FIR_TB.Unit \
    -start                 {} \
    -end                   {} \
    -block                 {} \
                           ../design/ncsim.vcd
```

What do these options mean?

`report_missing_nets` Any nets in the design that are not annotated by VCD are reported to this file. This is especially useful when pipe cleaning the flow. The default is false.

`Format` Voltus can read various activity files: VCD, TCF, SAF and FSDB.

`Scope` This is the name of the module within the activity file associated with the block or other part of the design that you want to analyze.

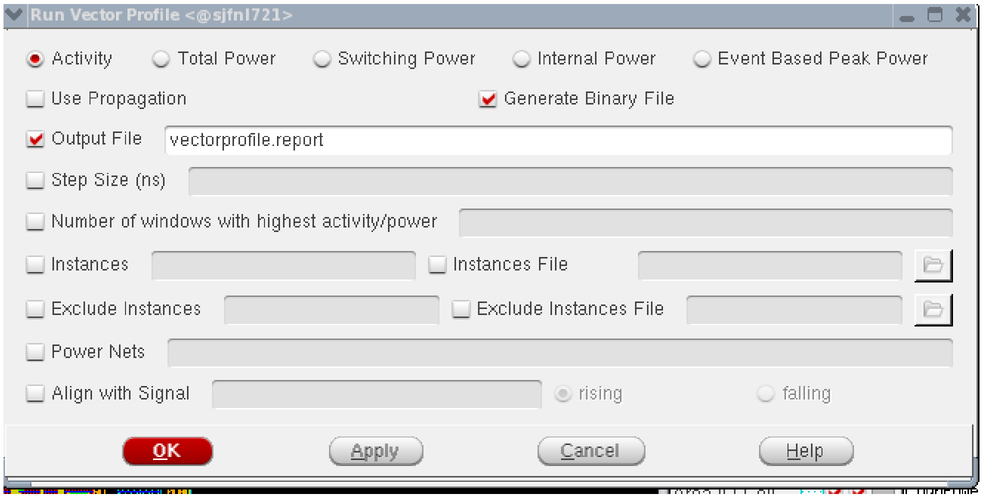
`Start` This is the point where the simulation will begin.

`End` This is the point where the simulation ends.

`Block` This is the name of the VCD block to map a sub-block activity file with a top level verilog file. This provides the ability to support multiple activity files based on the block instance names.

However, it is not advisable to perform power/rail analysis with the VCD for the entire simulation. Recommendation is to simulate only two dominant cycles worth of data. To compute worst case IR drop, you probably want to simulate two cycles with a lot of activity. The question is: how do you identify where the majority of the activity is taking place? The answer is the vector profiler.

9. In the *Run Power Analysis* window, click the *Vector Profiler* button. The following window will appear:



Run Vector Profile <@sjfnl721>

● Activity ○ Total Power ○ Switching Power ○ Internal Power ○ Event Based Peak Power

☐ Use Propagation ☑ Generate Binary File

☑ Output File vectorprofile.report

☐ Step Size (ns)

☐ Number of windows with highest activity/power

☐ Instances ☐ Instances File

☐ Exclude Instances ☐ Exclude Instances File

☐ Power Nets

☐ Align with Signal ○ rising ○ falling

10. TCL command:

```
report_vector_profile
```

For a list of options, refer to the documentation or type in the Voltus console:

```
report_vector_profile -help
```

11. Common options let you control:

`activity` By default, the profiler will analyze the VCD and identify windows with the highest activity.

`average_power` It can also look for windows with the highest total average power, switching power, or internal power.

`event_based_peak_power` This performs event-based vector profiling. This method computes power profile of every event on each net. The event-based peak power profiling enables you to accurately capture vectors that could produce peak power using very small resolution (1ps) and with better performance than average power profiling, which uses larger resolution to compute average toggle density.

`Step` This lets you vary the step size. By default, it uses the dominant clock period. In this case, the dominant clock is 62.5 MHz, and the step size is 32ns. For peak power profiling, the step size is configured as 0.01 ns.

`Stop` This is the point where the simulation ends.

`nworst` This is the number of worst intervals: default is 10, you can specify any number you want.

`detailed report` This will generate a detailed report for entire simulation time.

`write_profiling_db` This will write out profiling database, which can be viewed with SimVision.

`write_profiling_fsdb` This writes out the activity and power profiling databases in the FSDB format.

12. To begin, generate a report with the default options by pressing the "*Apply*" button. When the tools is running, be sure to note the following in the log file:

User did not set the step size. The step size is being configured as 16 ns:

```
  Filename (activity)                    : ../design/ncsim.vcd
  Names in file that matched to design   : 4464/4472
  Annotation coverage for this file      : 3165/3165 = 100%
```

It is critical to review the log and note the annotation coverage. If no nets are annotated, you might be referencing the wrong VCD file, or using the scope that may be incorrectly defined.

Conversely, if some nets are annotated but many more nets are not, it is possible that the VCD contains nets inside the standard cells. The bottom line is to review the log file and understand any mismatches.

13. Now see the results. The report file generated is called vectorprofile.report:

```
The 10 intervals have the maximum activity:
               0.020    to              16.020       1574
             432.020    to             448.020       1459
             576.020    to             592.020       1368
             560.020    to             576.020       1367
             688.020    to             704.020       1365
             672.020    to             688.020       1361
             464.020    to             480.020       1348
             624.020    to             640.020       1348
             544.020    to             560.020       1345
             528.020    to             544.020       1327
```

   a. The report lists the top ten most active windows. These windows are based on the dominant clock period (in this case 16ns).
   b. You can see that from the time 0.020ns to 16.020ns, there were 1574 transitions.
   c. The command to run the profiler is:

```
report_vector_profile \
    -activity \
    -write_profiling_db   true \
    -outfile              vectorprofile.report
```

14. Peak power profiling:

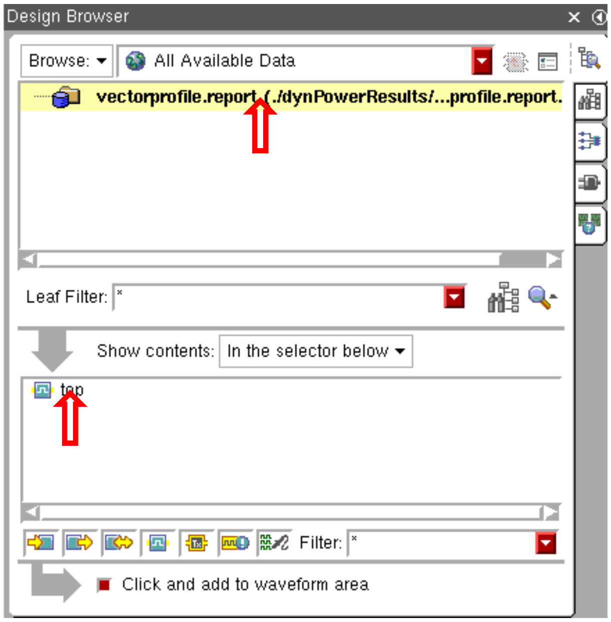
```
report_vector_profile \
      -event_based_peak_power \
      -write_profiling_db      true \
      -outfile                 vectorprofile_peak_power.report
```

See the report file, "`vectorprofile_peak_power.report`", which lists top 10 windows with the peak power number:

```
…
The 10 intervals have the maximum power (mW):
             0.170    to              0.180      570.4
             0.250    to              0.260      546.1
             4.780    to              4.790      530.9
            12.780    to             12.790      530.9
            20.780    to             20.790      530.9
            28.780    to             28.790      530.9
            36.780    to             36.790      530.9
            44.780    to             44.790      530.9
            52.780    to             52.790      530.9
            60.780    to             60.790      530.9
```

15. We can also use the waveform viewer to visualize the activity of the design. You might have noticed that we told Voltus to create a binary database of the profile data (.dsn and .trn files are written in the run directory). To use the waveform viewer:

    a. Open *Power & Rail -> Dynamic Results ->Waveforms*.
    b. In the "*View Waveforms*" window, select "*Profiling Histograms*".
    c. In the "*Waveforms Files*" section, click the "*Add*" button.
    d. Select the file "*vectorprofile.report.trn*" and press open.
    e. In the "*View Waveforms*" window, press "*Plot*".
    f. *SimVision* opens, but no waveform is displayed.
    g. In the Design Browser part of the window:
        i. Click on the "vectorprofile.report".
        ii. "top" will appear in the window below. Click on "top", and the waveform will appear:



Design Browser

Browse: All Available Data

vectorprofile.report (../dynPowerResults/...profile.report.

Leaf Filter: *

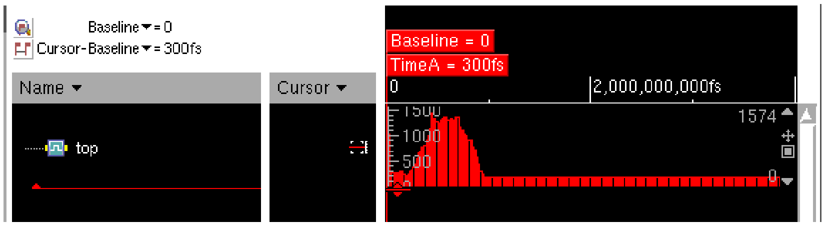
Show contents: In the selector below

top

Filter: *

Click and add to waveform area

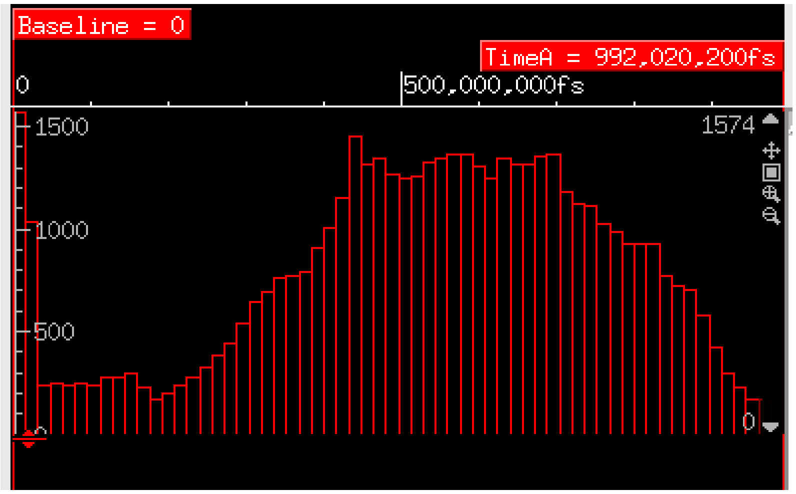
iii. Finally, *View->Zoom->Full X* and *Full Y* to see the full waveform:



Baseline = 0

Cursor-Baseline = 300fs

iv. Ultimately, you should be able to zoom in and view the activity over time:



v. The command to open the waveform viewer is:

```
    view_dynamic_waveform \
        -type              profile \
        -waveform_files  {vectorprofile.report.trn}
```

16. From the results, run the following command to base the simulation on the window 432-448.

```
    read_activity_file \
        -format        VCD \
        -scope         FIR_TB.Unit \
        -start         432 \
        -end           448 \
        -block         {} \
                       ../design/ncsim.vcd
    report_power \
        -outfile       super_filter.rpt
```

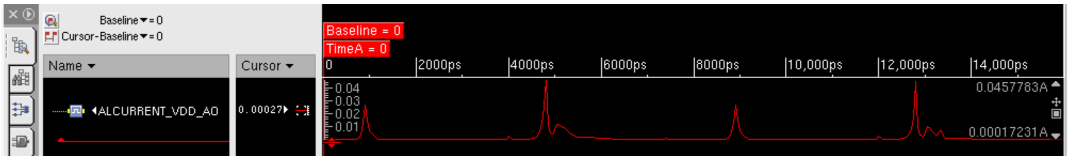
Or

Execute it from the GUI:

a. Open the *Run Power Analysis* window, add the *start* and *stop* times to the VCD file.
b. Be sure to press the “*Add*” button to add it to the stimulus list.
c. Also, select the previous stimulus (without the start/stop times) and press “*Remove*” button.
d. Press *OK* to start the simulation/power analysis.

17. At this point, the flow is the same as the vectorless flow for the current waveforms and drive rail analysis:

```
    view_dynamic_waveform \
  -type                      current \
  -composite_waveform_type  total_current \
  -waveform_files            {dynPowerResults/dynamic_VDD_AO.ptiavg}
```



Note: the total simulation is 16ns.

18. Do not exit Voltus.

Dynamic Rail Analysis

In this section, you will use the results from the dynamic power analysis to drive dynamic rail analysis. You can use either the vectorless or vector-based results. In this lab, we will use the vectorless results. However, feel free to experiment with the vector based results.

1. Set the rail analysis mode:

```
set_rail_analysis_mode \
    -method                  dynamic \
    -analysis_view           AV_wc_on \
    -power_switch_eco        true \
    -generate_movies         true \
    -generate_decap_eco      true \
    -decap_opt_method        feasibility \
    -save_voltage_waveforms  true \
    -accuracy                hd \
    -temperature             125 \
    -gif_resolution          0 \
    -power_grid_library { \
        ../data/pgv_dir/tech_pgv/techonly.cl \
        ../data/pgv_dir/stdcell_pgv/stdcells.cl \
        ../data/pgv_dir/macro_pgv/macros_pll.cl \
    }
```

What has stayed the same?

`power_grid_library` This is the path to the libraries to be used during analysis. Note that when the libraries were characterized, the directories had a .cl extension.

`Accuracy` xd This uses early view for all cells (fast performance).

hd This uses IR view for macros and EM view for EM check.

`analysis_view` This is used by Voltus to determine which power domains are on or off.

What has changed?

`Method` This is now dynamic.

What has been added?

`power_switch_eco` This creates eco file for power switch optimization.

`generate_movies` This generates movie; default is false.

Note: This option will impact performance in runtime and output data size, it is not recommended for a large design; please use it with caution.

`generate_decap_eco` This creates eco file for decap optimization.

`decap_opt_method` This defines method of decap optimization.

`save_voltage_waveforms` This saves the instance voltage waveform, the default is false.

`gif_resolution` This is a setting to generate suitably scaled GIF files to view movie.

2. Define the power pad location files. This is the same as used in the static rail analysis:

```
set_power_pads –reset
set_power_pads \
    -net     VDD_AO \
    -format  xy \
    -file    ../design/super_filter_VDD_AO.pp

set_power_pads \
    -net     VDD_external
    -format  xy \
    -file    ../design/super_filter_VDD_external.pp

set_power_pads \
    -net     VSS \
    -format  xy \
    -file    ../design/super_filter_VSS.pp
```

3. Now point to the current files created during dynamic power analysis:

```
set_power_data -reset
set_power_data \
    -format current { \
        dynVecLessPowerResults/dynamic_VDD_AO.ptiavg \
        dynVecLessPowerResults/dynamic_VDD_column.ptiavg \
        dynVecLessPowerResults/dynamic_VDD_ring.ptiavg \
        dynVecLessPowerResults/dynamic_VDD_external.ptiavg \
        dynVecLessPowerResults/dynamic_VSS.ptiavg \
    }
```

4. Finally, run rail analysis on all domains. The same as before; however, now we are pointing to a different results directory:

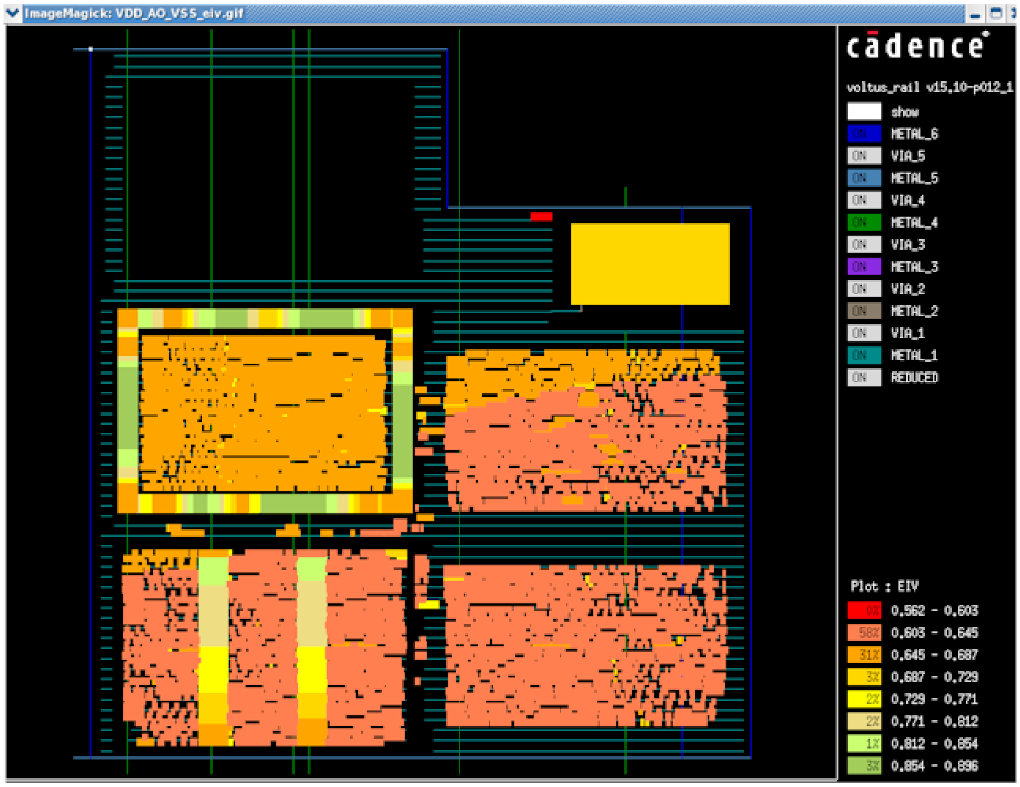
```
analyze_rail \
    -results_directory  ./dynVecLessRailResults \
    -type               domain \
                        ALL
```

5. It will take three minutes to complete rail analysis with 6CPUs with movie option. It is much faster without the movie option. The output data size is much smaller without generating movies.

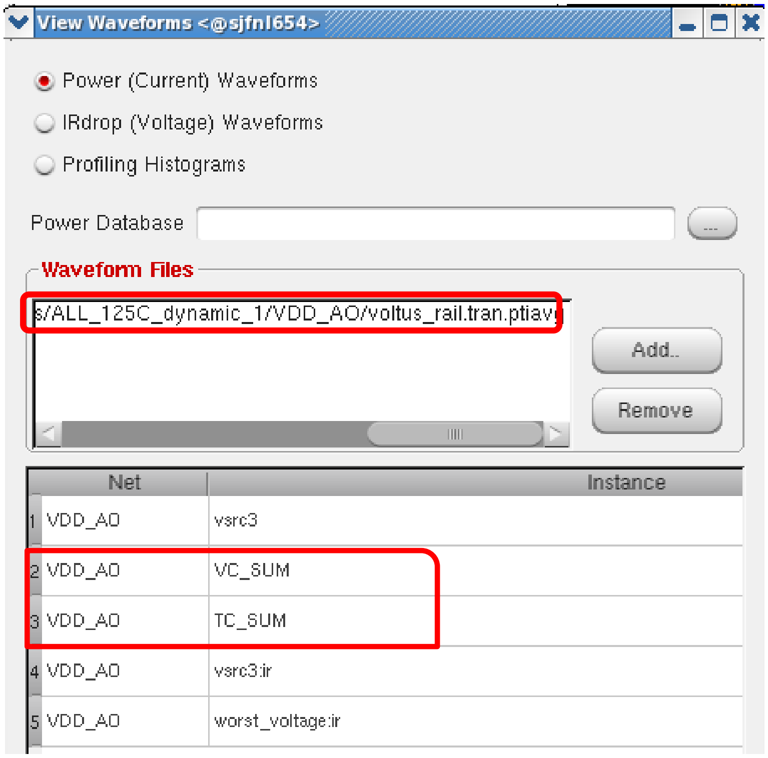
6. Once complete, like in static analysis, you can generate various plots (IR drop, resistor current, tap current, etc.).

   But what do these plots mean in the context of a dynamic simulation? These plots highlight the worst of the worst. For example, the IR drop plot shows the largest IR drop at each node throughout the simulation. Similarly, the resistor current plot shows the maximum current at any node throughout the simulation. If you are interested, feel free to open the Power & Rail Results window and review these plots.

7. Effective instance voltage data and plots: in output directory you can find plots of VDD_AO_VSS_eiv.gif and VDD_external_VSS_eiv.gif. You can also find effective instance voltage report files: VDD_AO_VSS.iv and VDD_external_VSS.iv. Following is the effective instance voltage for VDD_AO_VSS, which shows the effective operating voltage for each instance (the lowest is 0.562V):



8. TC_SUM and VC_SUM plots: Bring out *View Waveforms* form, and fill in *voltus_rail.tran.ptiavg* file from the output directory:



● Power (Current) Waveforms
○ IRdrop (Voltage) Waveforms
○ Profiling Histograms

Power Database

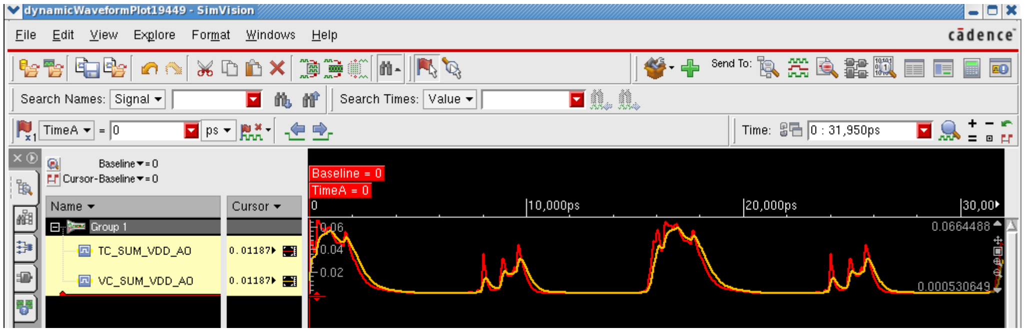
Waveform Files

s/ALL_125C_dynamic_1/VDD_AO/voltus_rail.tran.ptiav

	Net	Instance
1	VDD_AO	vsrc3
2	VDD_AO	VC_SUM
3	VDD_AO	TC_SUM
4	VDD_AO	vsrc3:ir
5	VDD_AO	worst_voltage:ir

The following are waveforms of *TC_SUM* and *VC_SUM* of net *VDD_AO*: you can see that VC_SUM is lower in magnitude and smoother that TC_SUM. This is due to local decap.

Feel free to view TC_SUM and VC_SUM waveforms for other power nets: VSS and VDD_external.



dynamicWaveformPlot19449 - SimVision

Search Names: Signal

Search Times: Value

TimeA = 0 ps

Time: 0 : 31,950ps

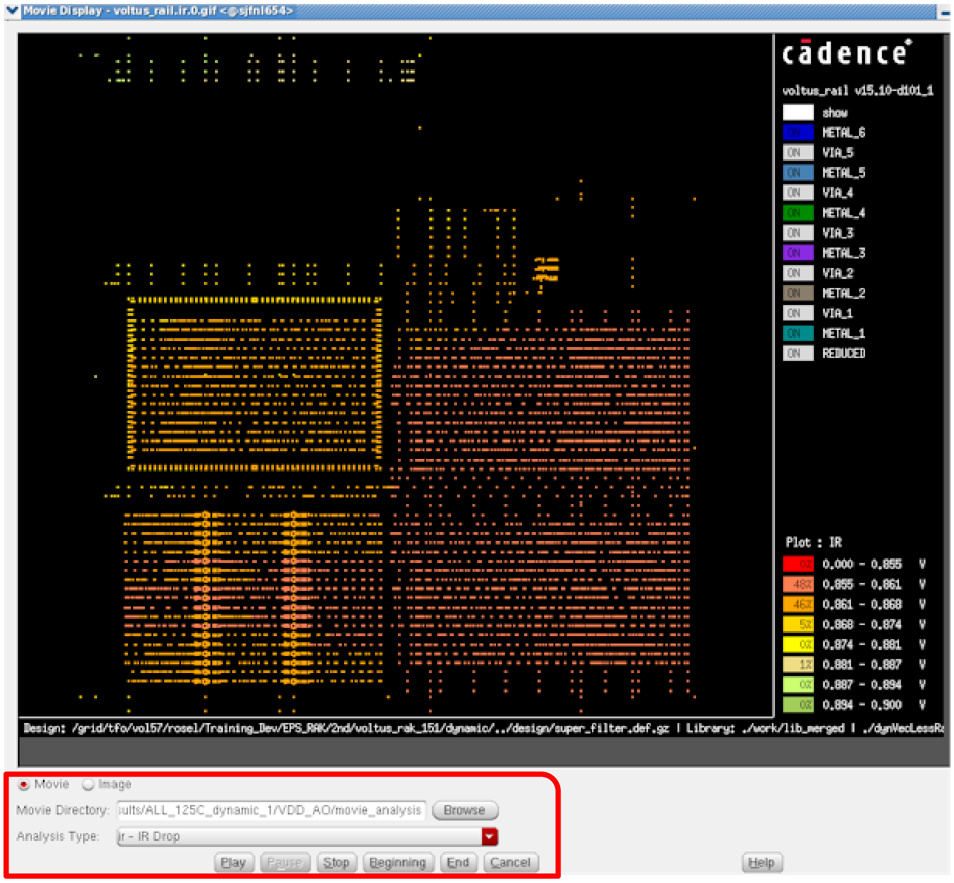
Baseline = 0

Cursor-Baseline = 0

Name	Cursor
Group 1	
TC_SUM_VDD_AO	0.01187
VC_SUM_VDD_AO	0.01187

9. You might find the movies that show IR drop and tap current over the course of the simulation to be more interesting. To watch either of these movies:

   a. Select Power & Rail->Dynamic Results-> Movies. The following window will appear:



Movie Display - voltus_rail.ir.0.gif <@sjfnl654>

voltus_rail v15.10-d101_1

Plot : IR

0% 0.000 - 0.855 V
48% 0.855 - 0.861 V
46% 0.861 - 0.868 V
5% 0.868 - 0.874 V
0% 0.874 - 0.881 V
1% 0.881 - 0.887 V
0% 0.887 - 0.894 V
0% 0.894 - 0.900 V

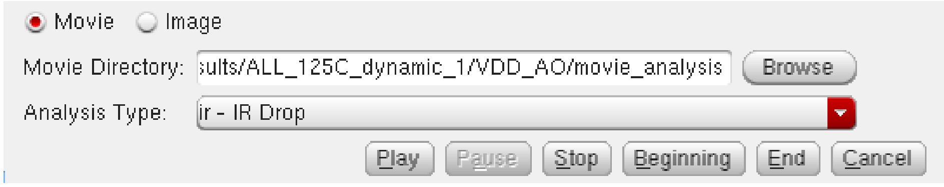
Design: /grid/tfo/vol57/rosel/Training_Dev/EPS_RAK/2nd/voltus_rak_151/dynamic/../design/super_filter.def.gz | Library: ./work/lib_merged | ./dynVecLessR

● Movie ○ Image

Movie Directory: sults/ALL_125C_dynamic_1/VDD_AO/movie_analysis

Analysis Type: ir - IR Drop

b. Click the “Browse” button. A new window will appear asking you to select a movie directory.
c. Navigate to dynVecLessRailResults/ALL_125C_dynamic_1/VDD_AO/movie_analysis directory and click“choose”.



● Movie ○ Image

Movie Directory: sults/ALL_125C_dynamic_1/VDD_AO/movie_analysis

Analysis Type: ir - IR Drop

d. In the View Movies window, select the Analysis Type to be “IR Drop”.
Press the “Play” button. The “Movie Display” window will appear and begin playing the IR drop movie.
e. The command to execute this on the command line is:

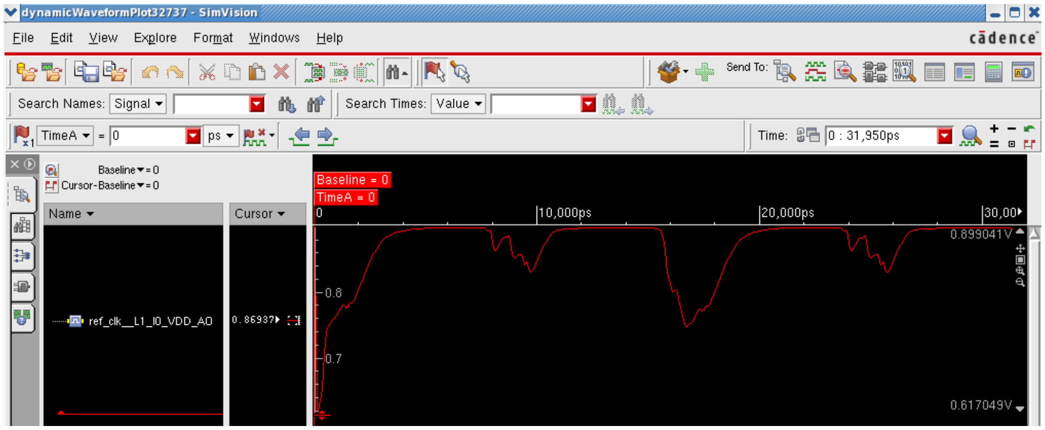
```
view_dynamic_movie \
  -type              ir \
  -movies_directory \
    dynVecLessRailResults/ALL_125C_dynamic_1/VDD_AO/movie_analysis
```

f. At any point, you can stop the movie and change the analysis type to Tap Current. This shows the current requirements of the instances over the duration of the simulation.

10. Additionally, you can view the voltage waveforms at any instance in the design with the waveform viewer. Here is how:

a. Select Power & Rail->Report->Dynamic Waveforms.
b. Select IR drop (voltage) waveforms.
c. Select the State Directory to be:
dynVecLessRailResults/ALL_125C_dynamic_1/VDD_AO
d. The VDD_AO.ptiavg waveform file will automatically load, and all of the instances will be displayed in the instance window.
e. Enter the instance ref_clk__L1_IO for the “Filter Pattern”.
f. Click instances. Now that instance will be the only one selected.
g. Highlight that instance and press “Plot”. The SimVision viewer will appear.
h. In the waveform viewer, select View->Zoom->Full Y. this will let you see the voltage of that instance over time.
i. The command to execute this on the command line is:

```
view_dynamic_waveform \
    -type             voltage \
    -instance_name    ref_clk__L1_I0 \
    -waveform_files { \
        dynVecLessRailResults/ALL_125C_dynamic_1/VDD_AO/VDD_AO.ptiavg \
    }
```



Feel free to explore other plots, report files, waveforms.

Native Power up Analysis

For details about native power up analysis please refer to the Voltus User Guide.

We will introduce native power up analysis with the following runs:

- Native power up power analysis
- Nets-based native power up rail analysis
- Block-based native power up rail analysis
- What-if capacitance analysis

1. Go to native_pu directory:

```
cd native_pu
```

2. Start Voltus and load the design:

```
voltus –init ../tcl/load_design.tcl
```

3. Run native power up power calculation by setting the “–native_powerup true” option, and the rest is the same as in vectorless dynamic power analysis:

```
set_power_analysis_mode \
        -method                         dynamic_vectorless \
        -disable_static                 true \
        -analysis_view                  AV_wc_on \
        -create_binary_db               false \
        -write_static_currents          false \
        -honor_negative_energy          true \
        -ignore_control_signals         true \
            -power_grid_library { \
            ../data/pgv_dir/tech_pgv/techonly.cl \
            ../data/pgv_dir/stdcell_pgv/stdcells.cl \
            ../data/pgv_dir/macro_pgv/macros_pll.cl \
                                        }
set_power_output_dir \
        -reset

set_power_output_dir \
        ./dynamic_pwr

set_default_switching_activity \
        -reset

set_default_switching_activity \
        -input_activity                 0.2 \
        -period                         10.0

set_powerup_analysis \
        -reset
```

```
set_powerup_analysis \
        -native_powerup                  true

report_power
```

4. View dynamic power output files:

In the output directory, you will see two additional output files, *pg_trigger_VDD_column* and *pg_trigger_VDD_ring*, which contain the arrival (rise/fall) times at the enable pin of the power-gate cells along with the transition times (rise/fall) for all the power-gate cells in the design. This information is further used by the Rail Analysis engine downstream to do rush current analysis.

Following is a portion of file *pg_trigger_VDD_ring*:

```
*                                                                                                   **
**                               Format of power-up Sequence file                                    **
**                                                                                                   **
** Power-gate_instance_name    Enable_Pin_name    rise_time    rise_slew    fall_time    fall_slew **
**                                                                                                   **
** Time unit is in seconds.                                                                          **
** Fall Time and Fall Slew are optional.                                                             **
**                                                                                                   **

ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_32_133_155 NSLEEPIN 3.35772e-08 9.3625e-11 4.60926e-08 9.95e-11
ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_150_4 NSLEEPIN 1.3808e-09 9.3625e-11 1.6277e-09 9.95e-11
ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_147_3 NSLEEPIN 1.1634e-09 9.3625e-11 1.3274e-09 9.95e-11
ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_145_2 NSLEEPIN 9.46e-10 9.3625e-11 1.0271e-09 9.95e-11
ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_142_1 NSLEEPIN 7.505e-10 5e-12 7.505e-10 5e-12
…
```

5. Run powering up analysis for the block(s) of interest:

This will take into account the parasitic of the always-on net, which is assuming a non-ideal always-on net. This flow lets you analyze more than one powerup net at a time. Therefore, you can simultaneously analyze the IR drop on the always-on net, and power up one or more switched nets in the same run.

You can run powering up analysis for *VDD_ring* with the following script:

```
set_rail_analysis_mode \
set_rail_analysis_mode \
        -method                          dynamic \
        -accuracy                        hd \
        -power_grid_library {\
            ../data/pgv_dir/tech_pgv/techonly.cl \
            ../data/pgv_dir/stdcell_pgv/stdcells.cl \
            ../data/pgv_dir/macro_pgv/macros_pll.cl } \
        -analysis_view                   AV_wc_on \
        -powering_up_rails               {VDD_ring } \
        -powerup_fast_mode               true \
        -vsrc_search_distance            50 \
        -generate_movies                 true \
        -save_voltage_waveforms true \
        -verbosity                      true

set_power_data \
        -reset

set_power_data \
        -format             current \
        -scale              1 \
                           {dynamic_pwr/dynamic_VDD_AO.ptiavg \
                            dynamic_pwr/dynamic_VDD_ring.ptiavg \
                            dynamic_pwr/dynamic_VDD_column.ptiavg \
                            dynamic_pwr/dynamic_VDD_external.ptiavg \
                            dynamic_pwr/dynamic_VSS.ptiavg}
set_power_pads \
        -reset

set_power_pads \
        -net                            VSS \
        -format                         xy \
        -file                           ../design/super_filter_VSS.pp

set_power_pads \
        -net                            VDD_external \
        -format                         xy \
        -file                     ../design/super_filter_VDD_external.pp

set_power_pads \
        -net                            VDD_AO \
        -format                         xy \
        -file                           ../design/super_filter_VDD_AO.pp

analyze_rail \
        -type                           domain \
        -results_directory              dynamic_rail_pu_nets \
                                        ALL
```

Because the movie option is on, the run time is about six minutes, which works with the Voltus 15.1isr1 and onward.

6. View native power up analysis result:

   a. You can see the following message in the output log file:

```
Power-up analysis results for net VDD_ring:

        Simulation time = 3.3608e-08s
        Threshold (Vt)  = 0.81V

        Measured maximum rush current          = 0.0245664A
        Measured wake-up time for switched net = 6.5e-10s

        Number of power switches turned on in this simulation = 152
[of total 152]
```

```
        Last power switch to turn-on in this simulation is
'ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_32_133_155:NSLEEPIN' at
time 3.3608e-08s.
```

b. There is a detailed report file in the rail analysis output directory for each powering up net. This summarizes max rush current value, simulation time, wake up time for switched net, total number of power switches, turn on time, and rush current for each power gate.

   See the following report file:

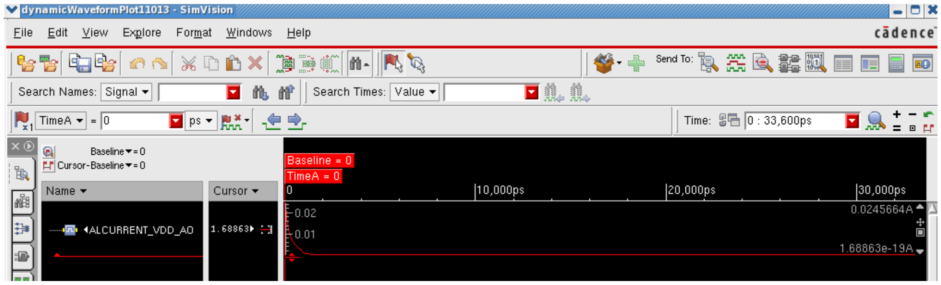
   dynamic_rail_pu_nets/ALL_25C_dynamic_1/VDD_ring/VDD_ring_powerup.report

```
Summary
=======

        Simulation time = 3.3608e-08s
        Threshold (Vt)  = 0.81V

        Measured maximum rush current          = 0.0245664A
        Measured wake-up time for switched net = 7.5e-10s

        Number of power switches turned on in this simulation =
152 [of total 152]
        Last power switch to turn-on in this simulation is
'ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_32_133_155:NSLEEPIN' at
time 3.3608e-08s.


Detailed Report
===============

   A. Reporting all the power gate instances which are turned on.

        ORDER       TURN-ON TIME  PEAK CURRENT    INSTANCES
        1             7.61026e-10   0.000161621
ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_142_1:NSLEEPIN
        2             9.76766e-10   0.000161621
ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_145_2:NSLEEPIN
        3             1.19417e-09   0.000161621
ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_147_3:NSLEEPIN
        4             1.41157e-09   0.000161621
ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_150_4:NSLEEPIN
…
```

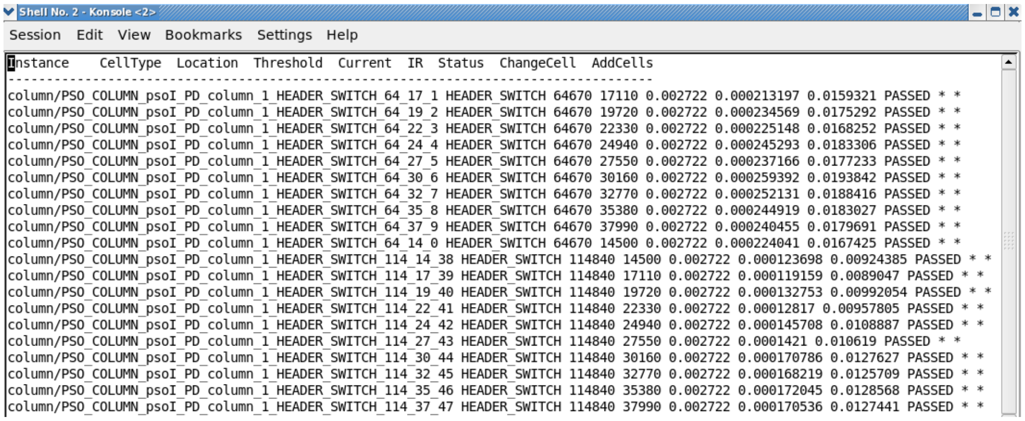
c. View the total rush current waveform using the TCL command:

```
view_dynamic_waveform \
    -type                     current \
    -waveform_files
dynamic_rail_pu_nets/ALL_25C_dynamic_1/VDD_ring/dynamic_powerup_c
urrent_VDD_ring_VDD_AO.ptiavg \
- instance_name            total_current_VDD_ring
```

Note: total_current_VDD_ring is a single entry available inside the "*dynamic_powerup_current_VDD_ring_VDD_AO.ptiavg*" file along with current waveforms for all power gate cells. If you use Composite Waveform option "*Total*", you will double the total current.



d. *pi.report*: a report file in the output directory reports the status of each power gate. Following is an example of the file:


Shell No. 2 - Konsole <2>
Session Edit View Bookmarks Settings Help

```
Instance    CellType  Location  Threshold  Current  IR  Status  ChangeCell  AddCells
-----------------------------------------------------------------------------------
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_64_17_1 HEADER_SWITCH 64670 17110 0.002722 0.000213197 0.0159321 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_64_19_2 HEADER_SWITCH 64670 19720 0.002722 0.000234569 0.0175292 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_64_22_3 HEADER_SWITCH 64670 22330 0.002722 0.000225148 0.0168252 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_64_24_4 HEADER_SWITCH 64670 24940 0.002722 0.000245293 0.0183306 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_64_27_5 HEADER_SWITCH 64670 27550 0.002722 0.000237166 0.0177233 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_64_30_6 HEADER_SWITCH 64670 30160 0.002722 0.000259392 0.0193842 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_64_32_7 HEADER_SWITCH 64670 32770 0.002722 0.000252131 0.0188416 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_64_35_8 HEADER_SWITCH 64670 35380 0.002722 0.000244919 0.0183027 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_64_37_9 HEADER_SWITCH 64670 37990 0.002722 0.000240455 0.0179691 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_64_14_0 HEADER_SWITCH 64670 14500 0.002722 0.000224041 0.0167425 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_114_14_38 HEADER_SWITCH 114840 14500 0.002722 0.000123698 0.00924385 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_114_17_39 HEADER_SWITCH 114840 17110 0.002722 0.000119159 0.0089047 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_114_19_40 HEADER_SWITCH 114840 19720 0.002722 0.000132753 0.00992054 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_114_22_41 HEADER_SWITCH 114840 22330 0.002722 0.00012817 0.00957805 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_114_24_42 HEADER_SWITCH 114840 24940 0.002722 0.000145708 0.0108887 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_114_27_43 HEADER_SWITCH 114840 27550 0.002722 0.0001421 0.010619 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_114_30_44 HEADER_SWITCH 114840 30160 0.002722 0.000170786 0.0127627 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_114_32_45 HEADER_SWITCH 114840 32770 0.002722 0.000168219 0.0125709 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_114_35_46 HEADER_SWITCH 114840 35380 0.002722 0.000172045 0.0128568 PASSED * *
column/PSO_COLUMN_psoI_PD_column_1_HEADER_SWITCH_114_37_47 HEADER_SWITCH 114840 37990 0.002722 0.000170536 0.0127441 PASSED * *
```

7. Run native power up analysis for block: This means powering up the block of interest without taking into account the always-on net, which is assuming an ideal always-on net. This flow lets you analyze only one powering up net at a time.

   Here is the detailed script:

```
set_rail_analysis_mode \
        -method                         dynamic \
        -accuracy                       hd \
        -power_grid_library             { \
            ../data/pgv_dir/tech_pgv/techonly.cl
            ../data/pgv_dir/stdcell_pgv/stdcells.cl \
            ../data/pgv_dir/macro_pgv/macros_pll.cl } \
        -analysis_view                  AV_wc_on \
        -block_powerup_rail             VDD_ring \
        -vsrc_search_distance           50

set_power_data \
        -reset

set_power_data \
        -format                     current \
        -scale                      1 \
                                    dynamic_pwr/dynamic_VDD_ring.ptiavg

set_power_pads \
        -reset

set_power_pads \
        -net                        VDD_AO \
        -format                     xy \
        -file                       ../design/super_filter_VDD_AO.pp

analyze_rail \
        -type                     net \
        -results_directory        ./dynamic_rail_block_VDD_ring_pu \
                                  VDD_ring
```

8. View rail analysis results:

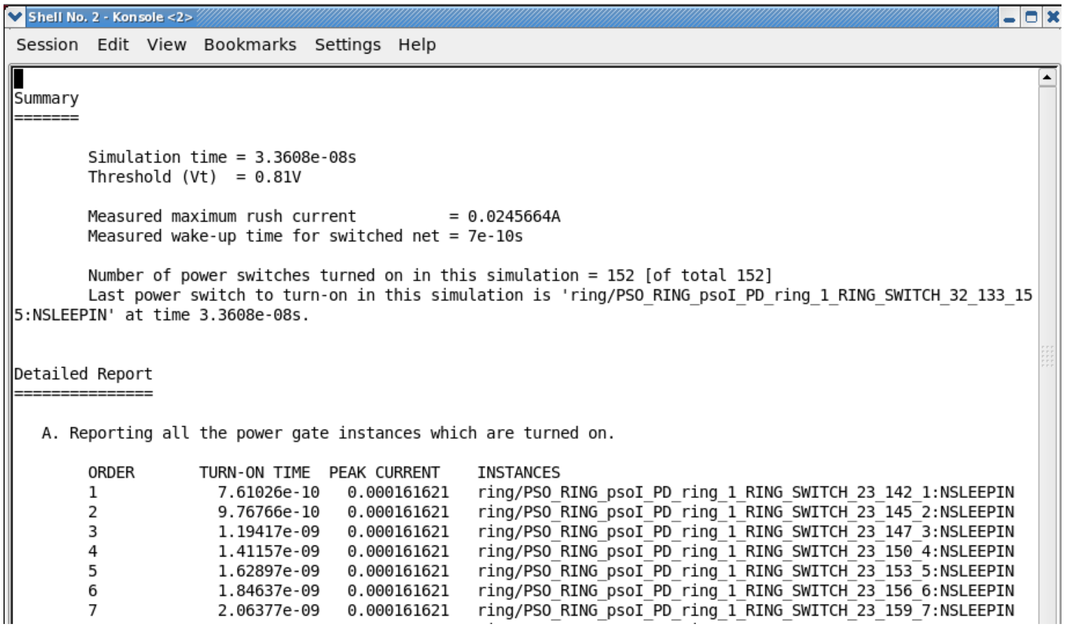
   a. The log file shows the following:

```
Power-up analysis results for net VDD_ring:

        Simulation time = 3.3608e-08s
        Threshold (Vt)  = 0.81V

        Measured maximum rush current          = 0.0245664A
        Measured wake-up time for switched net = 7e-10s

        Number of power switches turned on in this simulation = 152
[of total 152]
        Last power switch to turn-on in this simulation is
'ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_32_133_155:NSLEEPIN' at
time 3.3608e-08s.
```

   b. View the total rush current:

```
view_dynamic_waveform \
    -type                       current \
    -waveform_files             ./dynamic_rail_block_VDD_ring_pu/
VDD_ring_25C_dynamic_1/dynamic_powerup_VDD_ring_VDD_AO.ptiavg \
    - instance_name           total_current_VDD_ring
```

   c. View *VDD_ring_powerup.report* with the following summary:


Shell No. 2 - Konsole <2>
Session Edit View Bookmarks Settings Help

```
Summary
=======

        Simulation time = 3.3608e-08s
        Threshold (Vt)  = 0.81V

        Measured maximum rush current          = 0.0245664A
        Measured wake-up time for switched net = 7e-10s

        Number of power switches turned on in this simulation = 152 [of total 152]
        Last power switch to turn-on in this simulation is 'ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_32_133_15
5:NSLEEPIN' at time 3.3608e-08s.


Detailed Report
===============

   A. Reporting all the power gate instances which are turned on.

        ORDER        TURN-ON TIME  PEAK CURRENT    INSTANCES
        1              7.61026e-10   0.000161621   ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_142_1:NSLEEPIN
        2              9.76766e-10   0.000161621   ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_145_2:NSLEEPIN
        3              1.19417e-09   0.000161621   ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_147_3:NSLEEPIN
        4              1.41157e-09   0.000161621   ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_150_4:NSLEEPIN
        5              1.62897e-09   0.000161621   ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_153_5:NSLEEPIN
        6              1.84637e-09   0.000161621   ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_156_6:NSLEEPIN
        7              2.06377e-09   0.000161621   ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_23_159_7:NSLEEPIN
```

9. Run What-if capacitance analysis:

You can scale cap value to see the impact on the power up analysis result.

You scale cap value by 10X with the following command:

```
scale_what_if_capacitance \
        -global \
        -intrinsic_cap                 10 \
        -loading_cap                   10 \
        -grid_cap                      10
```

The complete script:

```
set_rail_analysis_mode \
        -method                        dynamic \
        -accuracy                      hd \
        -power_grid_library            { \

../data/pgv_dir/tech_pgv/techonly.cl \

../data/pgv_dir/stdcell_pgv/stdcells.cl \

../data/pgv_dir/macro_pgv/macros_pll.cl } \
        -analysis_view                 AV_wc_on \
        -block_powerup_rail            VDD_ring \
        -vsrc_search_distance          50

set_power_data \
        -reset

set_power_data \
        -format                         current \
        -scale                          1 \

dynamic_pwr/dynamic_VDD_ring.ptiavg

set_power_pads \
        -reset

set_power_pads \
        -net                            VDD_AO \
        -format                         xy \
        -file
../design/super_filter_VDD_AO.pp

scale_what_if_capacitance \
        -global \
        -intrinsic_cap                  10 \
        -loading_cap                    10 \
        -grid_cap                       10

analyze_rail \
        -type                           net \
        -results_directory
./dynamic_rail_block_VDD_ring_cap_scale VDD_ring
```

10. View capacitance impact on the power up analysis results.

The log file shows:

```
Power-up analysis results for net VDD_ring:

        Simulation time = 3.3608e-08s
        Threshold (Vt)  = 0.81V

        Measured maximum rush current          = 0.0245664A
        Measured wake-up time for switched net = 3.4e-09s

        Number of power switches turned on in this simulation = 152
[of total 152]
        Last power switch to turn-on in this simulation is
'ring/PSO_RING_psoI_PD_ring_1_RING_SWITCH_32_133_155:NSLEEPIN' at
time 3.3608e-08s.
```

Note: The wake up time is changed from 7.0e-10s to 3.4e-09s.

Load OA data base

Voltus can read OA data base.

1. cd to oa data base work directory:

```
cd oa
```

2. cp cds.lib over:

```
cp ../tcl/cds.lib .
```

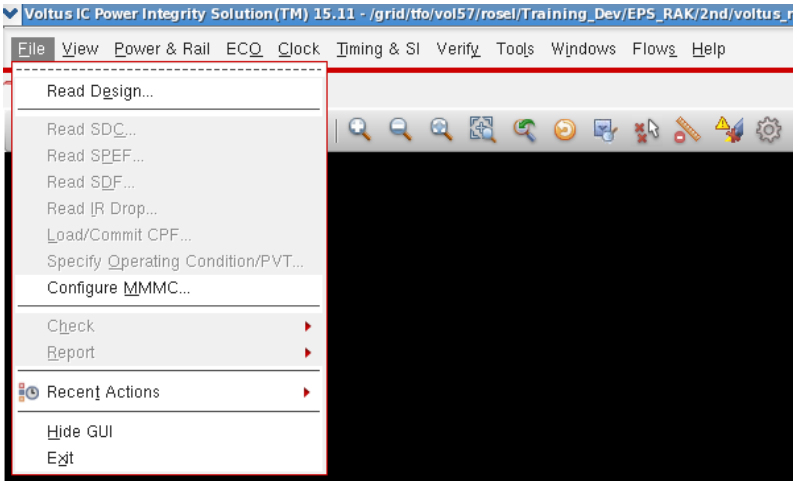
cds.lib defiles all the libraries:

```
SOFTINCLUDE
$(inst_root_with:share/cdssetup/cds.lib)/share/cdssetup/cds.lib

DEFINE gpdk090 ../data/oa_data/gpdk090

DEFINE gsclib090 ../data/oa_data/gsclib090

DEFINE lp ../data/oa_data/lp

DEFINE pll ../data/oa_data/pll

DEFINE decap ../data/oa_data/decap

DEFINE rakLib ../data/oa_data/rakLib
```

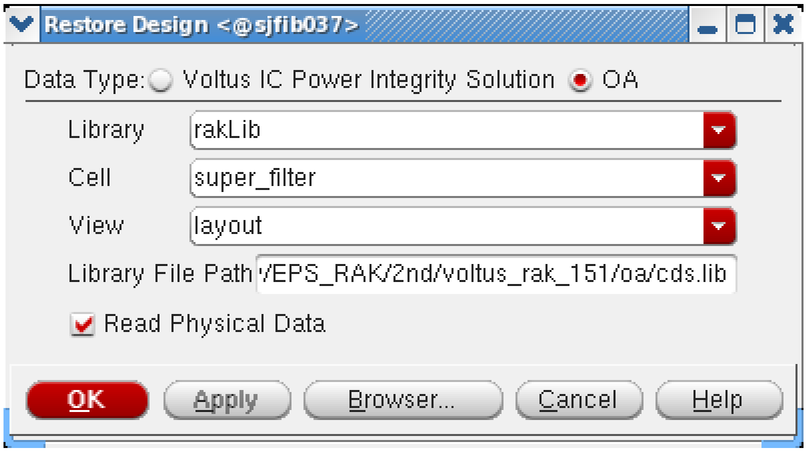
3. Start Voltus:

```
voltus
```

4. Select *File* -> *Read Design* to bring *Restore Design* window up:



5. File in Library name, Cell name and View name as the following and click *OK*:



Restore Design <@sjfib037>

Data Type: ○ Voltus IC Power Integrity Solution ● OA

Library rakLib

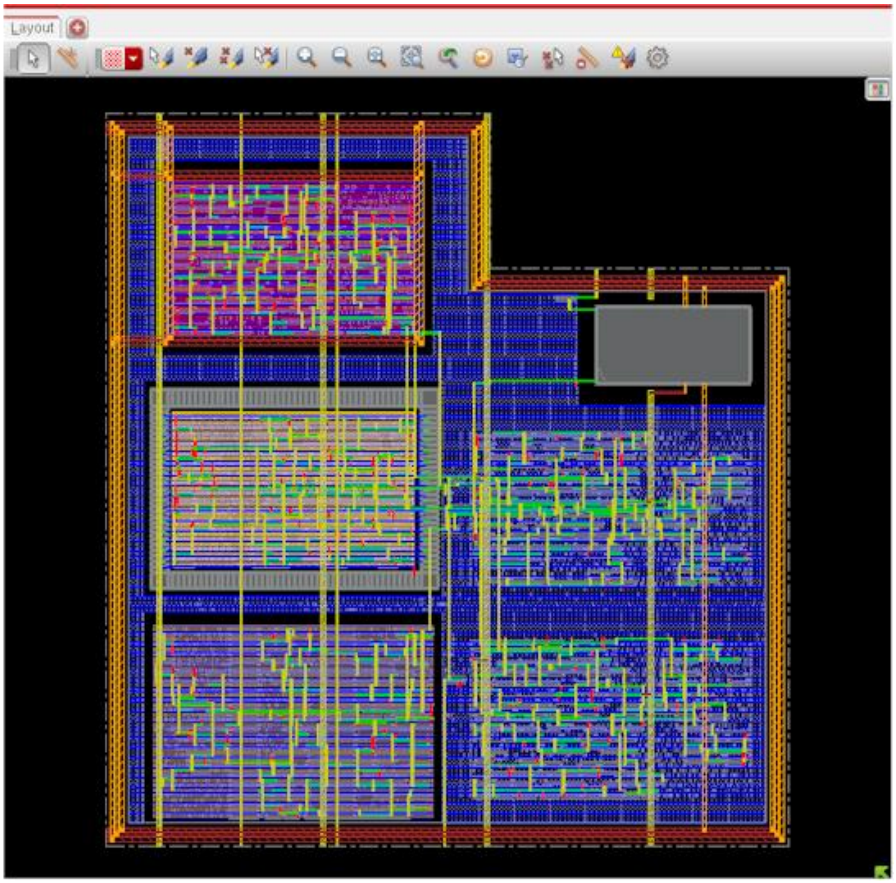
Cell super_filter

View layout

Library File Path /EPS_RAK/2nd/voltus_rak_151/oa/cds.lib

☑ Read Physical Data

Design will be loaded into Voltus:



Feel free to run some Voltus jobs as described in the previous sessions.

Early Rail Analysis (ERA)

ERA (early rail analysis) was a feature in Encounter, and is available in Voltus since Voltus 15.1 ERA.

1. Go to era directory:

   ```
   cd era
   ```

2. Start Voltus:

   ```
   voltus
   ```

3. Load the design:

   ```
   source ../tcl/load_design.tcl
   ```

4. Run ERA with the TCL command:

   ```
   source ../tcl/era.tcl
   ```

5. The TCL command file has the following:

   ```
   set_rail_analysis_mode \
       -method                       era_static \
       -accuracy                     xd \
       -extraction_tech_file         ../data/qrc/gpdk090_9l.tch \
       -temperature                  125 \
       -era_current_region_file      ../tcl/VSS_1.curRegion

   set_power_pads \
       -format                       xy \
       -file                         ../design/super_filter_VSS.pp\
       -net                          VSS

   set_pg_nets \
        -net                          VSS \
        -voltage                      0 \
        -threshold                    0.05
   analyze_rail \
           -type                      net \
                                      VSS
   ```

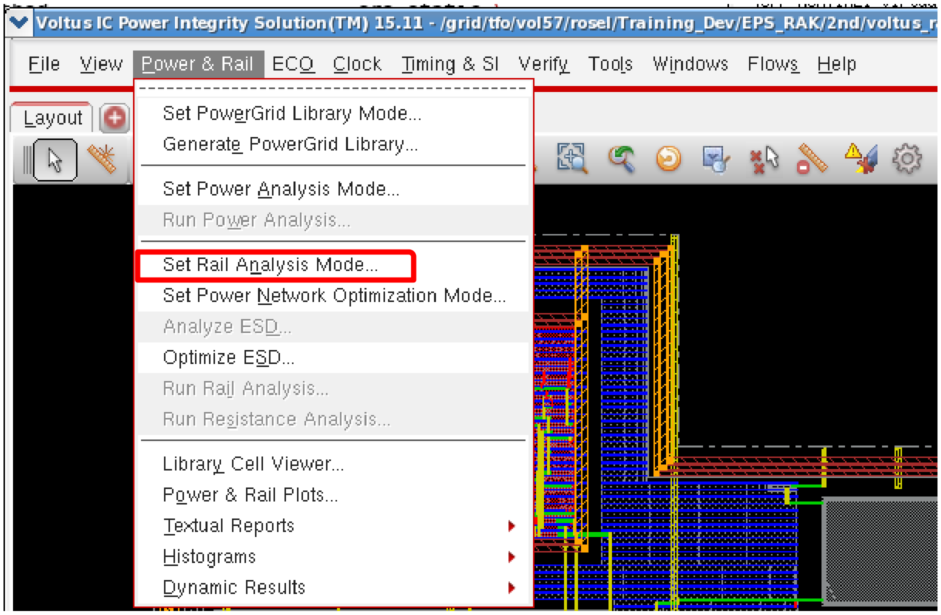
   - Make sure you use method "*era_static*".
   - You have option to use extraction (qrc) technology file without power-grid library. You can also use pgv library without the extraction technology file.
   - You need to provide *era_current_region_file.* Following is the format of the file:

     ```
     ##################################################

     #Format: LABEL name NET netname AREA x1 y1 x2 y2 LAYER layername
     <CURRENT value | PWL (t1 i1 t2 i2...)> INTRINSIC_CAP value
     LOADING_CAP value
     ```

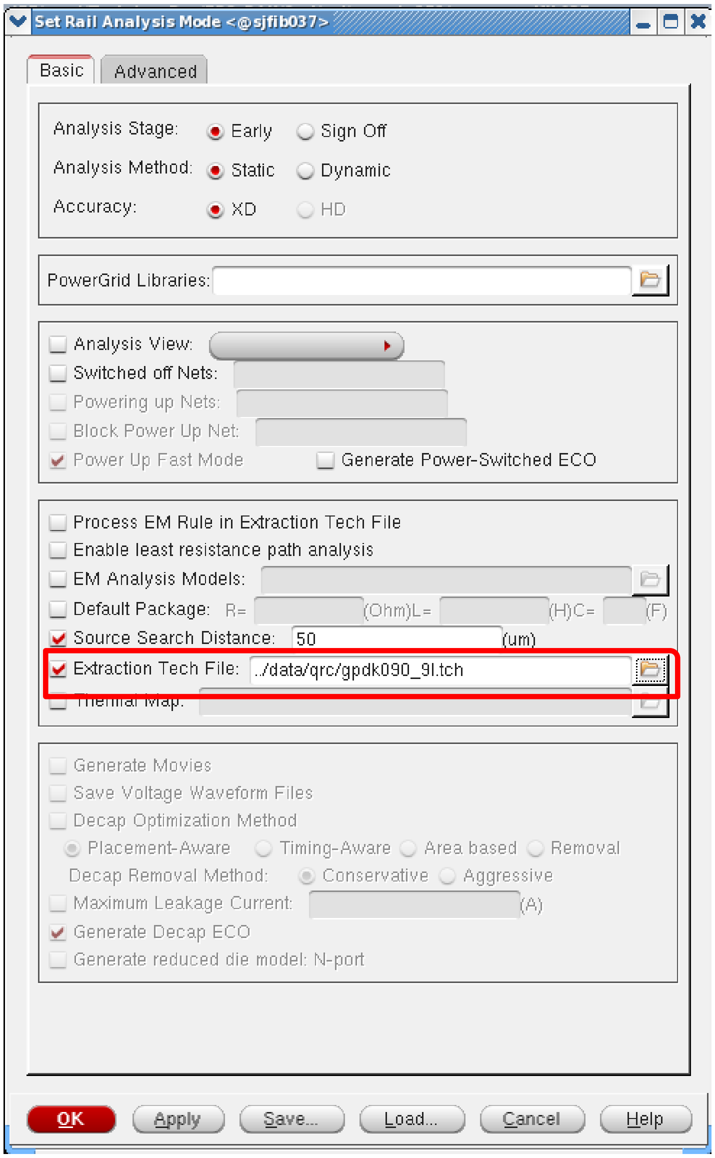
```
#Unit: current mA, cap pf, time ns, coordinate um

##################################################

label lab1 net VSS area 2.5505 0.3735 349.514 377.865 layer
Metal4 current 100
```

6. Run ERA through the GUI.

   a. Bring *Set Rail Analysis Mode* window up through: *Power & Rail -> Set Rail Analysis Mode*:


Voltus IC Power Integrity Solution(TM) 15.11 - /grid/tfo/vol57/rosel/Training_Dev/EPS_RAK/2nd/voltus_r
File
View
Power & Rail
ECO
Clock
Timing & SI
Verify
Tools
Windows
Flows
Help
Layout
Set PowerGrid Library Mode...
Generate PowerGrid Library...
Set Power Analysis Mode...
Run Power Analysis...
Set Rail Analysis Mode...
Set Power Network Optimization Mode...
Analyze ESD...
Optimize ESD...
Run Rail Analysis...
Run Resistance Analysis...
Library Cell Viewer...
Power & Rail Plots...
Textual Reports
Histograms
Dynamic Results

   b. Fill in the Basic tag as the following: use the extraction technology file and make sure to fill the location and file name in the form:



Set Rail Analysis Mode <@sjfib037>

Basic | Advanced

Analysis Stage: ● Early ○ Sign Off

Analysis Method: ● Static ○ Dynamic

Accuracy: ● XD ○ HD

PowerGrid Libraries:

□ Analysis View:

□ Switched off Nets:

□ Powering up Nets:

□ Block Power Up Net:

☑ Power Up Fast Mode □ Generate Power-Switched ECO

□ Process EM Rule in Extraction Tech File

□ Enable least resistance path analysis

□ EM Analysis Models:

□ Default Package: R= (Ohm)L= (H)C= (F)

☑ Source Search Distance: 50 (um)

☑ Extraction Tech File: ../data/qrc/gpdk090_9l.tch

□ Thermal Map:

□ Generate Movies

□ Save Voltage Waveform Files

□ Decap Optimization Method

● Placement-Aware ○ Timing-Aware ○ Area based ○ Removal

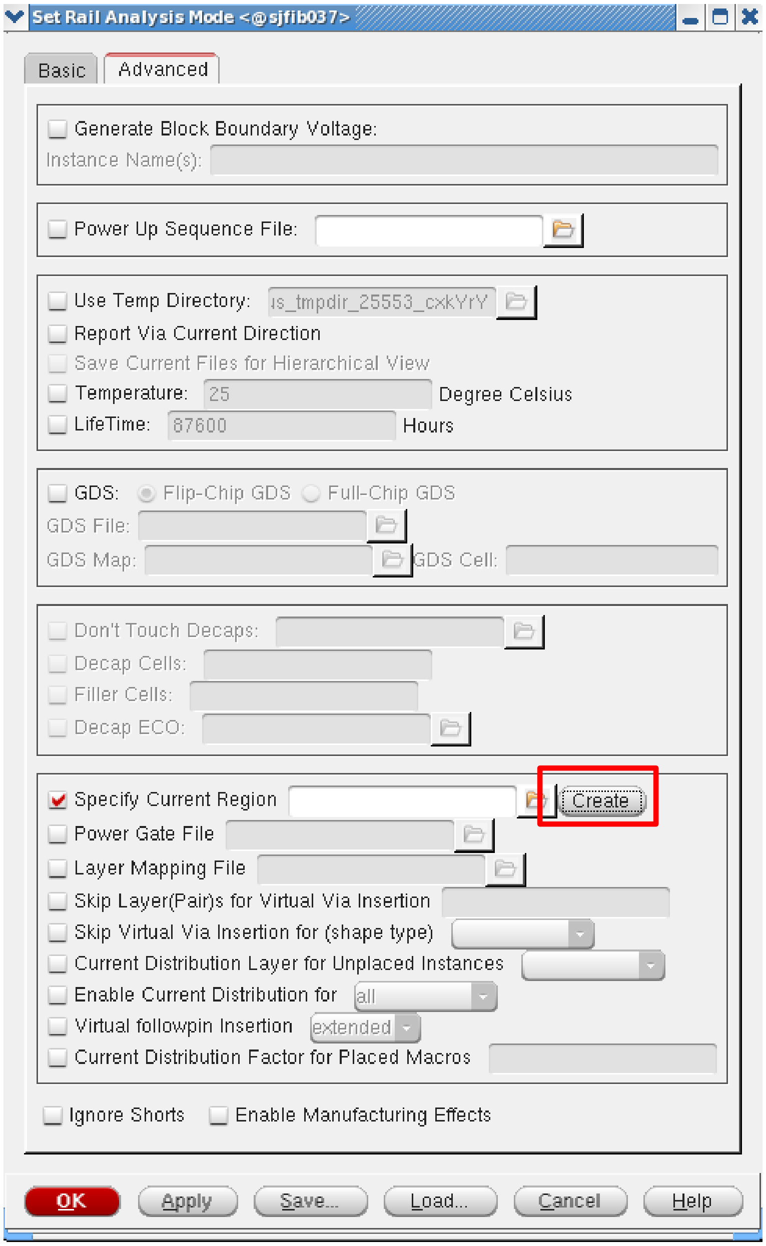
Decap Removal Method: ● Conservative ○ Aggressive

□ Maximum Leakage Current: (A)

☑ Generate Decap ECO

□ Generate reduced die model: N-port

c. Click Advanced tag, check *Specify Current Region*:



Set Rail Analysis Mode <@sjfib037>

Basic | Advanced

□ Generate Block Boundary Voltage:
Instance Name(s):

□ Power Up Sequence File:

□ Use Temp Directory: ıs_tmpdir_25553_cxkYrY
□ Report Via Current Direction
□ Save Current Files for Hierarchical View
□ Temperature: 25 Degree Celsius
□ LifeTime: 87600 Hours

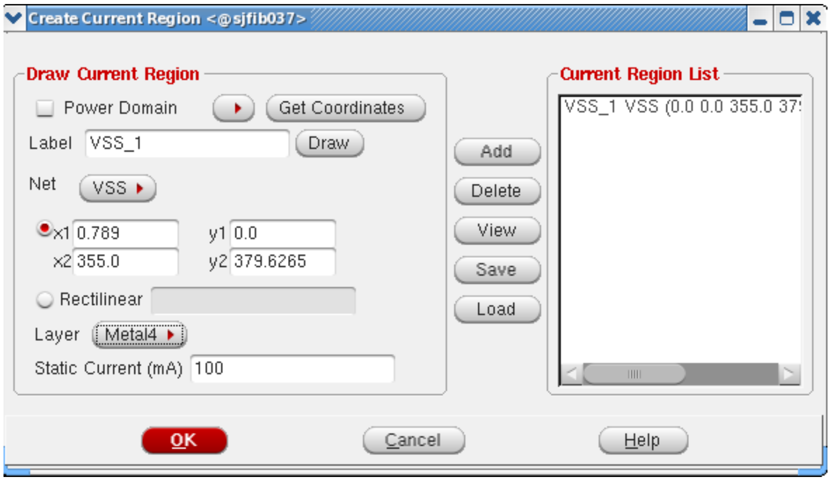
□ GDS: ● Flip-Chip GDS ○ Full-Chip GDS
GDS File:
GDS Map: GDS Cell:

□ Don't Touch Decaps:
□ Decap Cells:
□ Filler Cells:
□ Decap ECO:

☑ Specify Current Region [Create]
□ Power Gate File
□ Layer Mapping File
□ Skip Layer(Pair)s for Virtual Via Insertion
□ Skip Virtual Via Insertion for (shape type)
□ Current Distribution Layer for Unplaced Instances
□ Enable Current Distribution for all
□ Virtual followpin Insertion extended
□ Current Distribution Factor for Placed Macros

□ Ignore Shorts □ Enable Manufacturing Effects

d. Use the GUI to create current region file by clicking the *Create* button:



Create Current Region <@sjfib037>

Draw Current Region

☐ Power Domain

Label VSS_1

Net VSS

● x1 0.789 y1 0.0

x2 355.0 y2 379.6265

○ Rectilinear

Layer Metal4

Static Current (mA) 100

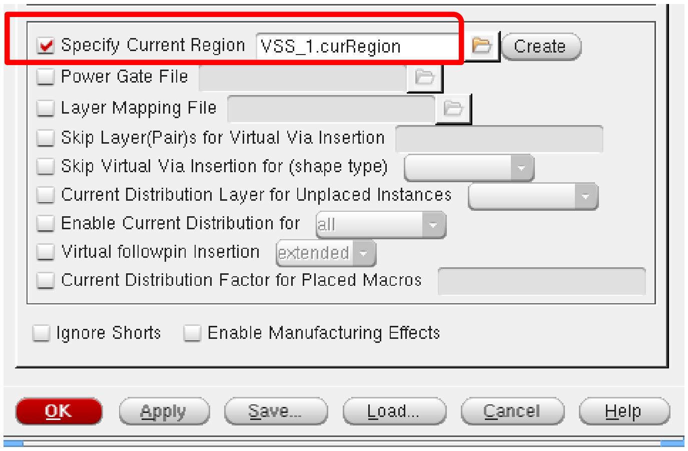
Current Region List

VSS_1 VSS (0.0 0.0 355.0 37!

- Select Net *VSS*.
- Select Current Layer *Metal4*.
- Enter *Static Current (mA)* as 100.
- Click *Draw*.
- Draw the whole area of the design.
- Click *Get Coordinates*.
- Click the Add button to add the current region to *Current Region List*.

You can create multiple current regions and make sure to add it to the *Current Region List.*

- Save it to a file: VSS_1.curRegion.
- Make sure the file name appears in the following field:



☑ Specify Current Region VSS_1.curRegion

☐ Power Gate File

☐ Layer Mapping File

☐ Skip Layer(Pair)s for Virtual Via Insertion

☐ Skip Virtual Via Insertion for (shape type)

☐ Current Distribution Layer for Unplaced Instances

☐ Enable Current Distribution for all

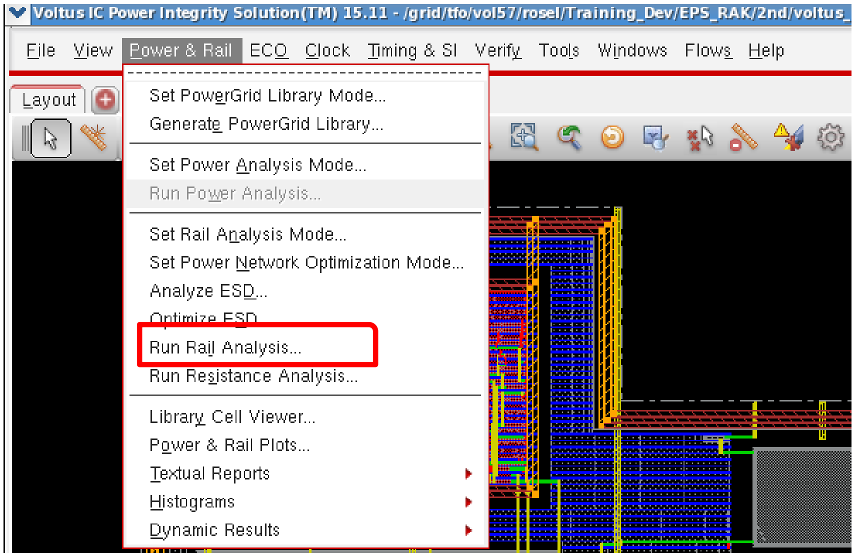
☐ Virtual followpin Insertion extended

☐ Current Distribution Factor for Placed Macros

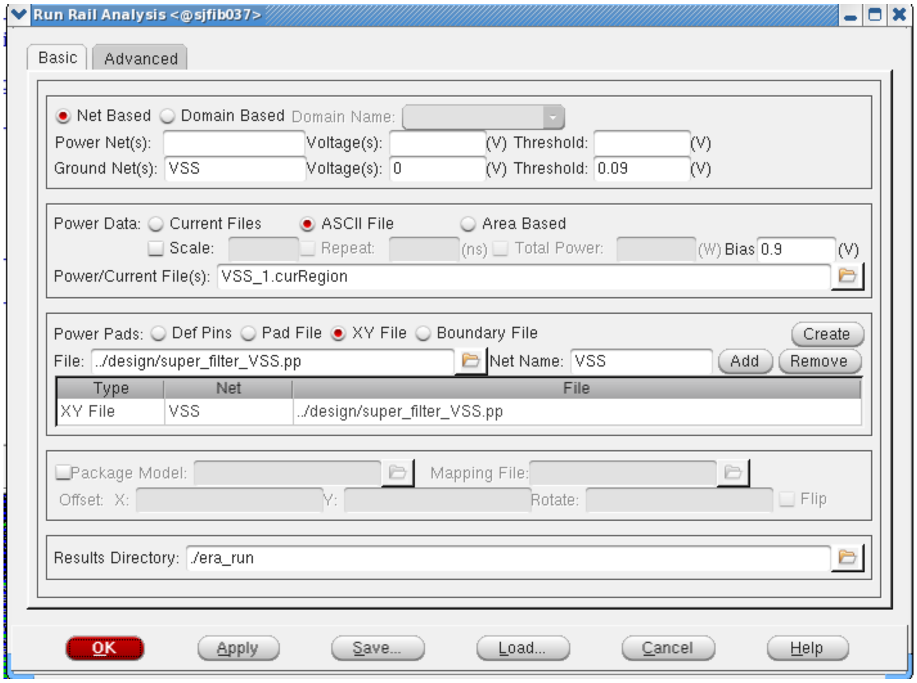
☐ Ignore Shorts ☐ Enable Manufacturing Effects

e. Click *OK* to close the window.

f. Bring up the run Rail Analysis form: *Rower & Rail -> Run Rail Analysis*:


Voltus IC Power Integrity Solution(TM) 15.11 - /grid/tfo/vol57/rosel/Training_Dev/EPS_RAK/2nd/voltus_
File View Power & Rail ECO Clock Timing & SI Verify Tools Windows Flows Help
Layout
Set PowerGrid Library Mode...
Generate PowerGrid Library...
Set Power Analysis Mode...
Run Power Analysis...
Set Rail Analysis Mode...
Set Power Network Optimization Mode...
Analyze ESD...
Optimize ESD
Run Rail Analysis...
Run Resistance Analysis...
Library Cell Viewer...
Power & Rail Plots...
Textual Reports
Histograms
Dynamic Results

g. fill in the window as the following:


Run Rail Analysis <@sjfib037>

Basic | Advanced

● Net Based ○ Domain Based Domain Name:

Power Net(s): Voltage(s): (V) Threshold: (V)

Ground Net(s): VSS Voltage(s): 0 (V) Threshold: 0.09 (V)

Power Data: ○ Current Files ● ASCII File ○ Area Based

□ Scale: □ Repeat: (ns) □ Total Power: (W) Bias 0.9 (V)

Power/Current File(s): VSS_1.curRegion

Power Pads: ○ Def Pins ○ Pad File ● XY File ○ Boundary File Create

File: ../design/super_filter_VSS.pp Net Name: VSS Add Remove

Type	Net	File
XY File	VSS	../design/super_filter_VSS.pp

□ Package Model: Mapping File:

Offset: X: Y: Rotate: □ Flip

Results Directory: ./era_run

OK Apply Save... Load... Cancel Help

h. Click *OK*.

i. After the job is done, feel free to see the analysis results.

j. Feel free to run ERA on other rails and domains.

Resistance Analysis

The Effective Resistance Analysis feature lets you perform effective resistance calculation for all the instances in a design in both net-based and domain-based analysis modes. When an instance has multiple pins, the effective resistance are computed at all the pins, and the one with the worst value is reported. If the instance uses the accurate view, the PGV resistance is taken into account during effective resistance calculation.

In this section, try the following resistance analysis for the following three cases:

- Domain-based instance to pad effective resistance analysis
- Net-based node to pad effective resistance analysis
- Net-based node to node effective resistance analysis

1. cd to res_analysis directory:

```
cd res_analysis
```

2. start Voltus:

```
voltus
```

3. Load the design:

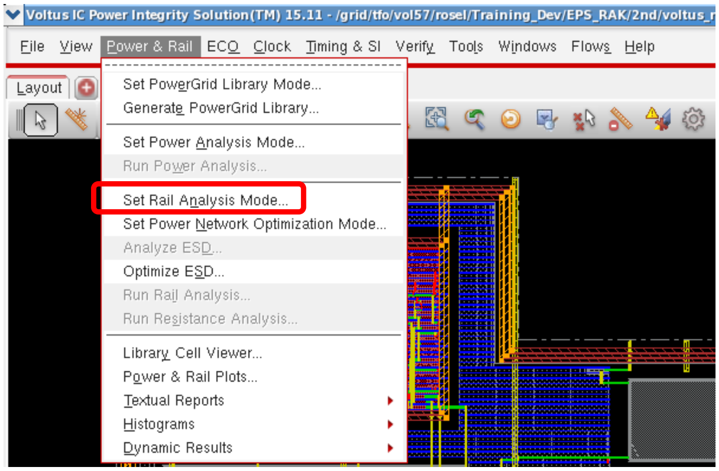
```
source ../tcl/load_design.tcl
```

4. Start the GUI:

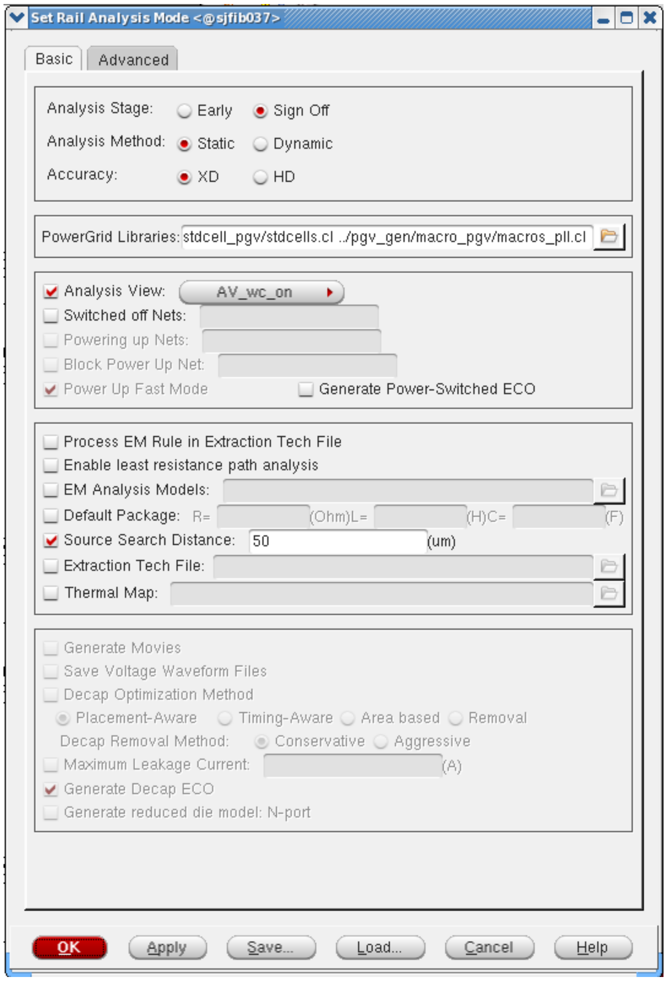
```
start_gui
```

5. Set the rail analysis mode:

   a. Bring *Set Rail Analysis Mode* window up through *Power & Rail -> Set Rail Analysis Mode*:


Voltus IC Power Integrity Solution(TM) 15.11 - /grid/tfo/vol57/rosel/Training_Dev/EPS_RAK/2nd/voltus_r
File
View
Power & Rail
ECO
Clock
Timing & SI
Verify
Tools
Windows
Flows
Help
Layout
Set PowerGrid Library Mode...
Generate PowerGrid Library...
Set Power Analysis Mode...
Run Power Analysis...
Set Rail Analysis Mode...
Set Power Network Optimization Mode...
Analyze ESD...
Optimize ESD...
Run Rail Analysis...
Run Resistance Analysis...
Library Cell Viewer...
Power & Rail Plots...
Textual Reports
Histograms
Dynamic Results

b. Fill the details as shown in the following image (same as static rail analysis):



Set Rail Analysis Mode <@sjfib037>

Basic | Advanced

Analysis Stage: ○ Early ● Sign Off

Analysis Method: ● Static ○ Dynamic

Accuracy: ● XD ○ HD

PowerGrid Libraries: stdcell_pgv/stdcells.cl ../pgv_gen/macro_pgv/macros_pll.cl

☑ Analysis View: AV_wc_on

□ Switched off Nets:

□ Powering up Nets:

□ Block Power Up Net:

☑ Power Up Fast Mode □ Generate Power-Switched ECO

□ Process EM Rule in Extraction Tech File

□ Enable least resistance path analysis

□ EM Analysis Models:

□ Default Package: R= (Ohm)L= (H)C= (F)

☑ Source Search Distance: 50 (um)

□ Extraction Tech File:

□ Thermal Map:

□ Generate Movies

□ Save Voltage Waveform Files

□ Decap Optimization Method

● Placement-Aware ○ Timing-Aware ○ Area based ○ Removal

Decap Removal Method: ● Conservative ○ Aggressive

□ Maximum Leakage Current: (A)

☑ Generate Decap ECO

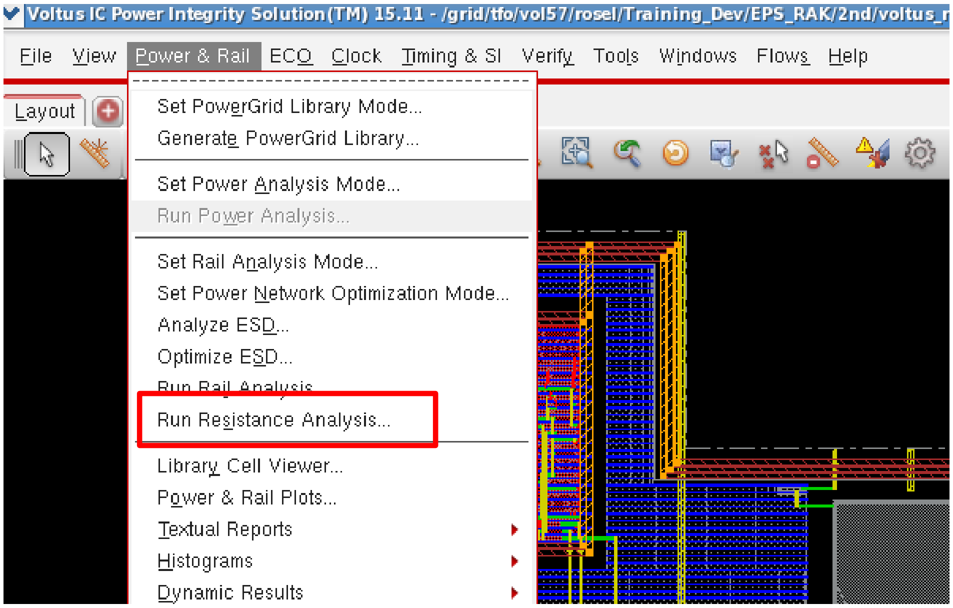
□ Generate reduced die model: N-port

c. TCL command (same as the static rail analysis mode setup):

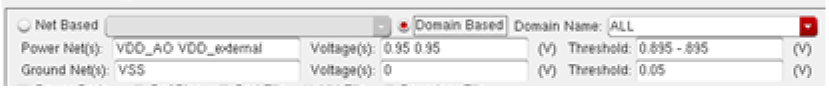
```
set_rail_analysis_mode \
    -method                                         static \
    -accuracy                                       xd \
    -analysis_view                                  AV_wc_on \
    -power_grid_library\
                { \
                ../data/pgv_dir/tech_pgv/techonly.cl \
                ../data/pgv_dir/stdcell_pgv/stdcells.cl \
                ../data/pgv_dir/macro_pgv/macros_pll.cl \
                } \
    -verbosity                                      true \
    -temperature                                    125
```

6. Setup domain based effective resistance analysis:

a. Bring up the *Run Resistance Analysis* window through *Power & Rail -> Run Resistance Analysis*:


Voltus IC Power Integrity Solution(TM) 15.11 - /grid/tfo/vol57/rosel/Training_Dev/EPS_RAK/2nd/voltus_r
File
View
Power & Rail
ECO
Clock
Timing & SI
Verify
Tools
Windows
Flows
Help
Layout
Set PowerGrid Library Mode...
Generate PowerGrid Library...
Set Power Analysis Mode...
Run Power Analysis...
Set Rail Analysis Mode...
Set Power Network Optimization Mode...
Analyze ESD...
Optimize ESD...
Run Rail Analysis...
Run Resistance Analysis...
Library Cell Viewer...
Power & Rail Plots...
Textual Reports
Histograms
Dynamic Results

b. Set up domain-based analysis:

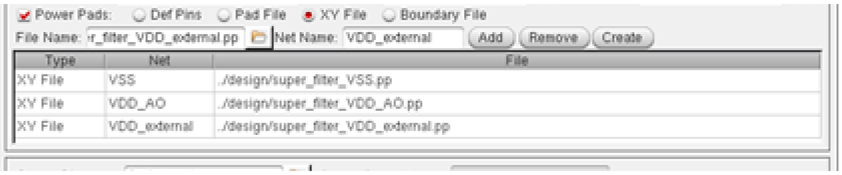


○ Net Based ● Domain Based Domain Name: ALL

Power Net(s): VDD_AO VDD_external Voltage(s): 0.95 0.95 (V) Threshold: 0.895 - .895 (V)

Ground Net(s): VSS Voltage(s): 0 (V) Threshold: 0.05 (V)

c. Set up the power pad location file:

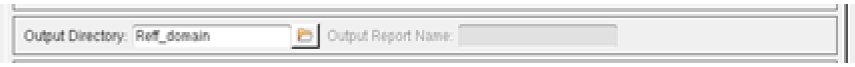


☑ Power Pads: ○ Def Pins ○ Pad File ● XY File ○ Boundary File

File Name: r_filter_VDD_external.pp Net Name: VDD_external Add Remove Create

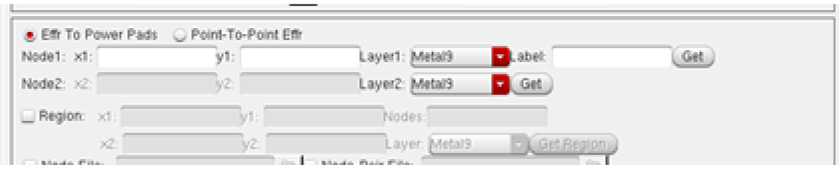
Type	Net	File
XY File	VSS	../design/super_filter_VSS.pp
XY File	VDD_AO	../design/super_filter_VDD_AO.pp
XY File	VDD_external	../design/super_filter_VDD_external.pp

d. Set up the output directory:



Output Directory: Reff_domain Output Report Name:

e. Make sure *Effr to Power Pads* option is checked:



● Effr To Power Pads ○ Point-To-Point Effr

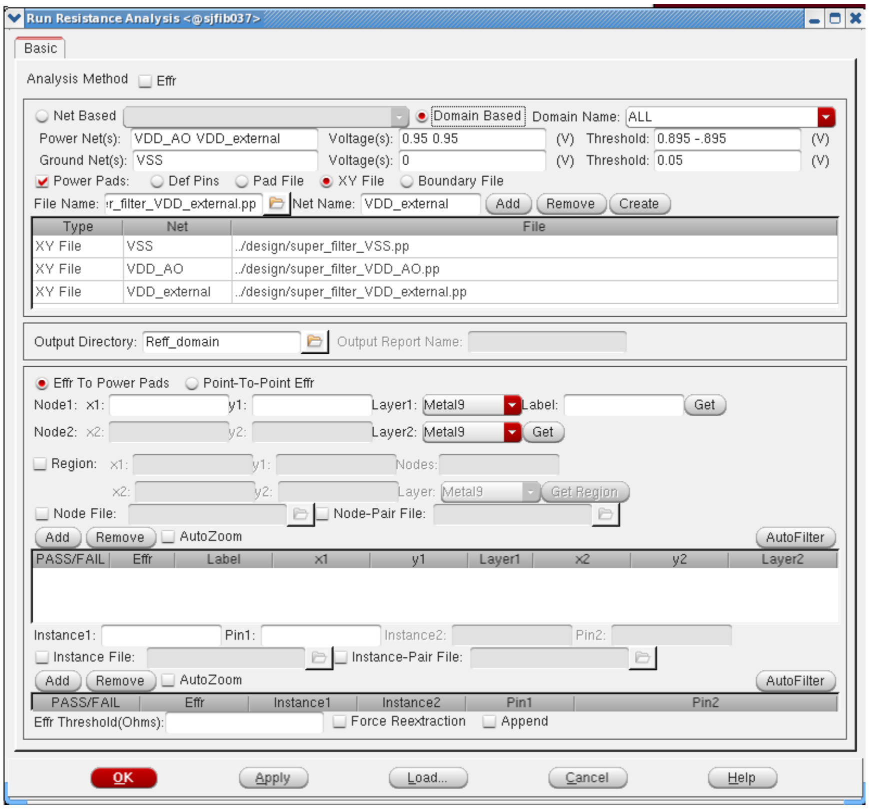
Node1: x1: y1: Layer1: Metal9 Label: Get

Node2: x2: y2: Layer2: Metal9 Get

☐ Region: x1: y1: Nodes:

x2: y2: Layer: Metal9 Get Region

f. Fill up the form as shown in the following image:



Run Resistance Analysis <@sjfib037>

Basic

Analysis Method □ Effr

○ Net Based ● Domain Based Domain Name: ALL

Power Net(s): VDD_AO VDD_external Voltage(s): 0.95 0.95 (V) Threshold: 0.895 -.895 (V)

Ground Net(s): VSS Voltage(s): 0 (V) Threshold: 0.05 (V)

☑ Power Pads: ○ Def Pins ○ Pad File ● XY File ○ Boundary File

File Name: r_filter_VDD_external.pp Net Name: VDD_external

Type	Net	File
XY File	VSS	../design/super_filter_VSS.pp
XY File	VDD_AO	../design/super_filter_VDD_AO.pp
XY File	VDD_external	../design/super_filter_VDD_external.pp

Output Directory: Reff_domain Output Report Name:

● Effr To Power Pads ○ Point-To-Point Effr

Node1: x1: y1: Layer1: Metal9 Label:

Node2: x2: y2: Layer2: Metal9

□ Region: x1: y1: Nodes:

x2: y2: Layer: Metal9

□ Node File: □ Node-Pair File:

□ AutoZoom

PASS/FAIL	Effr	Label	x1	y1	Layer1	x2	y2	Layer2

Instance1: Pin1: Instance2: Pin2:

□ Instance File: □ Instance-Pair File:

□ AutoZoom

PASS/FAIL	Effr	Instance1	Instance2	Pin1	Pin2

Effr Threshold(Ohms): □ Force Reextraction □ Append

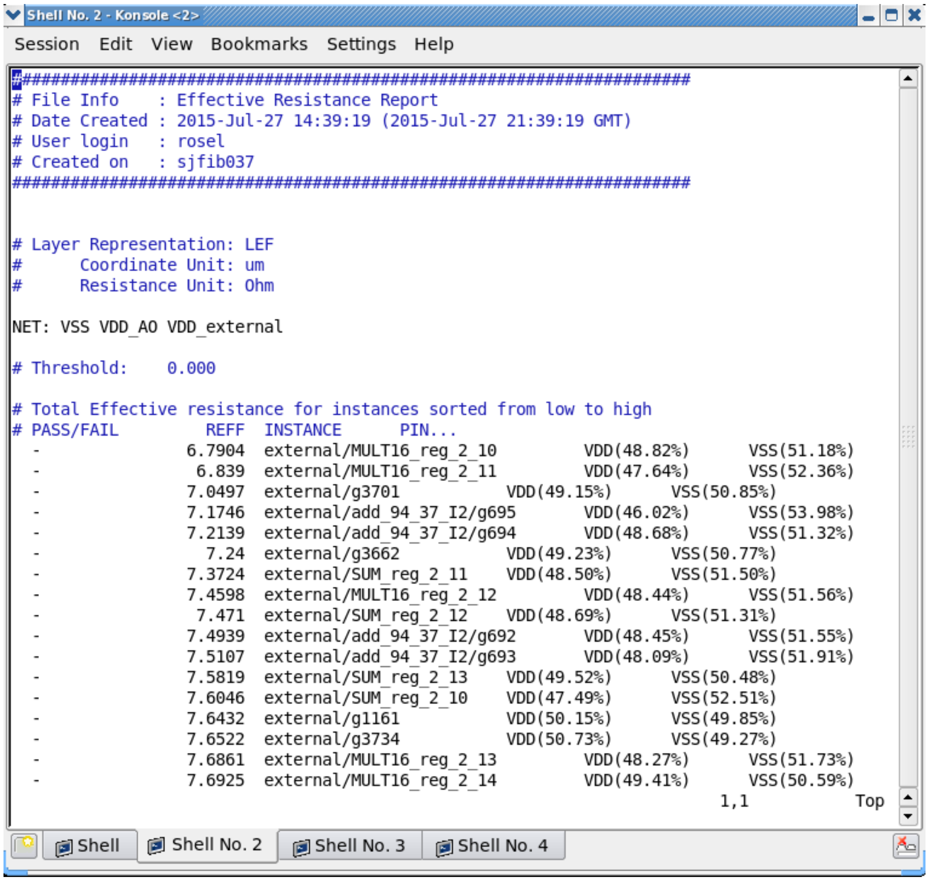
Note: Effective resistance analysis does not require power analysis.

g. Click *OK* to run.

h. TCL command:

```
set_power_pads \
        -reset

set_power_pads \
        -net                         VSS \
        -format                      xy \
        -file
../design/super_filter_VSS.pp

set_power_pads \
        -net                         VDD_external \
        -format                      xy \
        -file
../design/super_filter_VDD_external.pp

set_power_pads \
        -net                         VDD_AO \
        -format                      xy \
        -file
../design/super_filter_VDD_AO.pp

analyze_resistance \
        -domain                         ALL \
        -output_dir                     Reff_domain
```

i. View effective analysis result in text report file: “*Reff_domain/ALL_25C_reff_1/domain_effr.rpt*”



```
######################################################################
# File Info    : Effective Resistance Report
# Date Created : 2015-Jul-27 14:39:19 (2015-Jul-27 21:39:19 GMT)
# User login   : rosel
# Created on   : sjfib037
######################################################################


# Layer Representation: LEF
#      Coordinate Unit: um
#      Resistance Unit: Ohm

NET: VSS VDD_AO VDD_external

# Threshold:    0.000

# Total Effective resistance for instances sorted from low to high
# PASS/FAIL         REFF  INSTANCE      PIN...
  -               6.7904  external/MULT16_reg_2_10        VDD(48.82%)      VSS(51.18%)
  -                6.839  external/MULT16_reg_2_11        VDD(47.64%)      VSS(52.36%)
  -               7.0497  external/g3701          VDD(49.15%)      VSS(50.85%)
  -               7.1746  external/add_94_37_I2/g695      VDD(46.02%)      VSS(53.98%)
  -               7.2139  external/add_94_37_I2/g694      VDD(48.68%)      VSS(51.32%)
  -                 7.24  external/g3662          VDD(49.23%)      VSS(50.77%)
  -               7.3724  external/SUM_reg_2_11    VDD(48.50%)      VSS(51.50%)
  -               7.4598  external/MULT16_reg_2_12        VDD(48.44%)      VSS(51.56%)
  -                7.471  external/SUM_reg_2_12    VDD(48.69%)      VSS(51.31%)
  -               7.4939  external/add_94_37_I2/g692      VDD(48.45%)      VSS(51.55%)
  -               7.5107  external/add_94_37_I2/g693      VDD(48.09%)      VSS(51.91%)
  -               7.5819  external/SUM_reg_2_13    VDD(49.52%)      VSS(50.48%)
  -               7.6046  external/SUM_reg_2_10    VDD(47.49%)      VSS(52.51%)
  -               7.6432  external/g1161          VDD(50.15%)      VSS(49.85%)
  -               7.6522  external/g3734          VDD(50.73%)      VSS(49.27%)
  -               7.6861  external/MULT16_reg_2_13        VDD(48.27%)      VSS(51.73%)
  -               7.6925  external/MULT16_reg_2_14        VDD(49.41%)      VSS(50.59%)
                                                                        1,1           Top
```

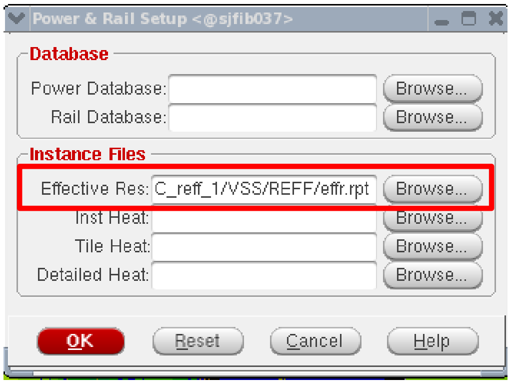
Shell No. 2 - Konsole <2>
Session Edit View Bookmarks Settings Help
Shell
Shell No. 2
Shell No. 3
Shell No. 4

The file reports the total effective resistance values (Rvdd + Rvss) for all the nets in the domain and the percentage of each net. There are also effective resistance report for each net available. For example, for VSS, the report file is:

Reff_domain/ALL_25C_reff_1/VSS/REFF/effr.rpt

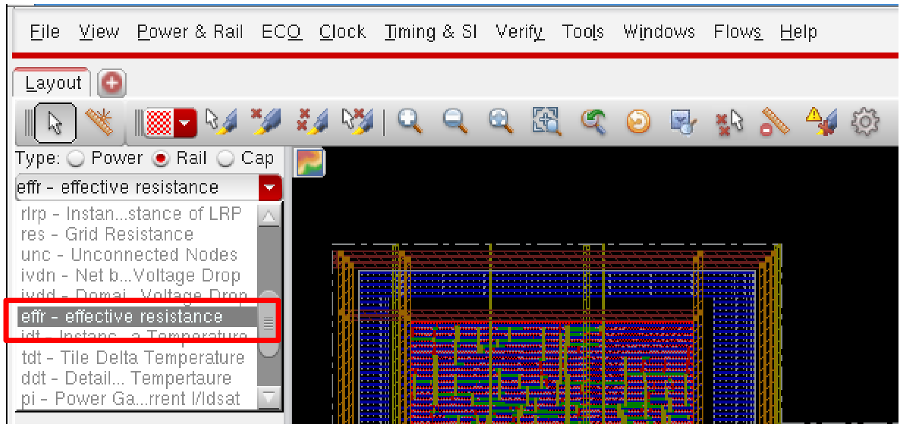
j. View effective analysis result in GUI:

You can load the text report file *effr.rpt* file into GUI to view, by specifying the report file into the *Power & Rail Setup* form.



Effective Res: C_reff_1/VSS/REFF/effr.rpt

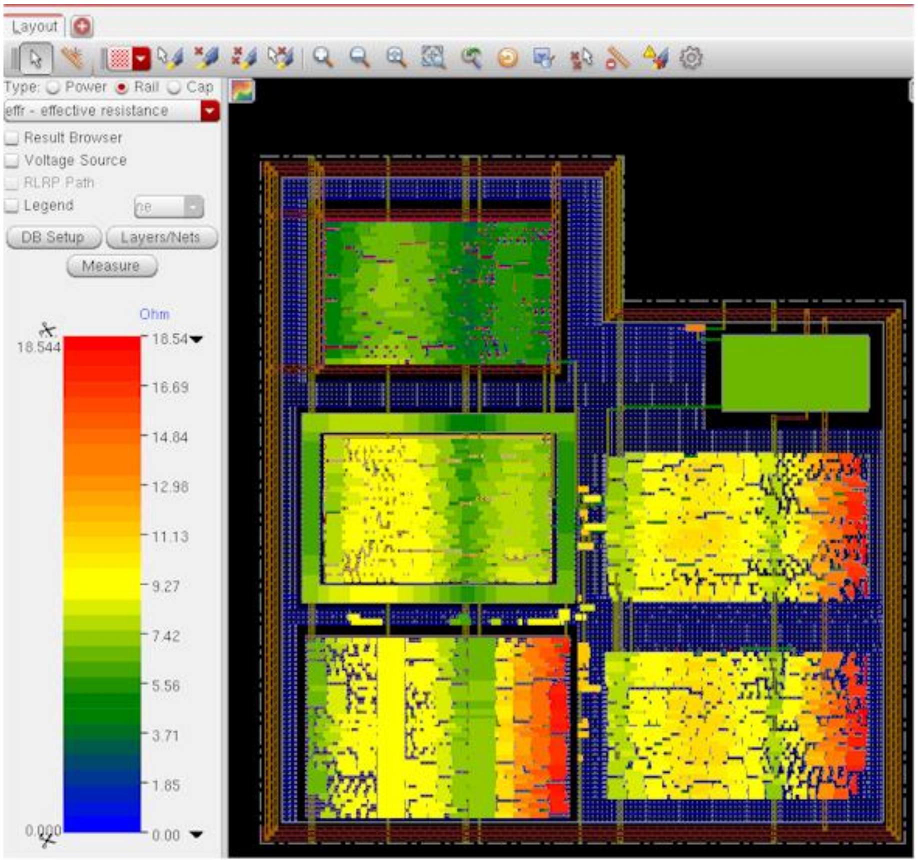
Then, select the plot type: *effr*:


File
View
Power & Rail
ECO
Clock
Timing & SI
Verify
Tools
Windows
Flows
Help
Layout
rlrp - Instan...stance of LRP
res - Grid Resistance
unc - Unconnected Nodes
ivdn - Net b...Voltage Drop
ivdd - Domai...Voltage Drop
effr - effective resistance
idt - Instanc...e Temperature
tdt - Tile Delta Temperature
ddt - Detail... Tempertaure
pi - Power Ga...rrent I/Idsat

Type: ○ Power ● Rail ○ Cap

effr - effective resistance

The VSS effective resistance plot appears as the following:



Type: ○ Power ● Rail ○ Cap

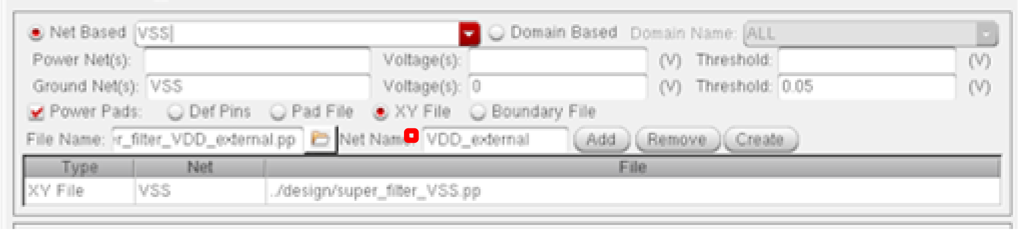
effr - effective resistance

Feel free to look for effr for net VDD_AO and net VDD_external.

7. Run net-based node to pad the effective resistance analysis:

   You do not need to re-set rail analysis mode but you do need to set *Run Resistance Analysis*:

   a. Bring up "*Run Resistance Analysis*" window.
   b. Set it to *Net based* and select *VSS*.
   c. Specify *VSS* pad location file name (remove VDD_AO and VDD_external location files using the *Remove* button):



● Net Based VSS ○ Domain Based Domain Name: ALL

Power Net(s): Voltage(s): (V) Threshold: (V)

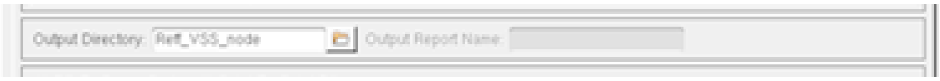
Ground Net(s): VSS Voltage(s): 0 (V) Threshold: 0.05 (V)

☑ Power Pads: ○ Def Pins ○ Pad File ● XY File ○ Boundary File

File Name: r_filter_VDD_external.pp Net Name VDD_external

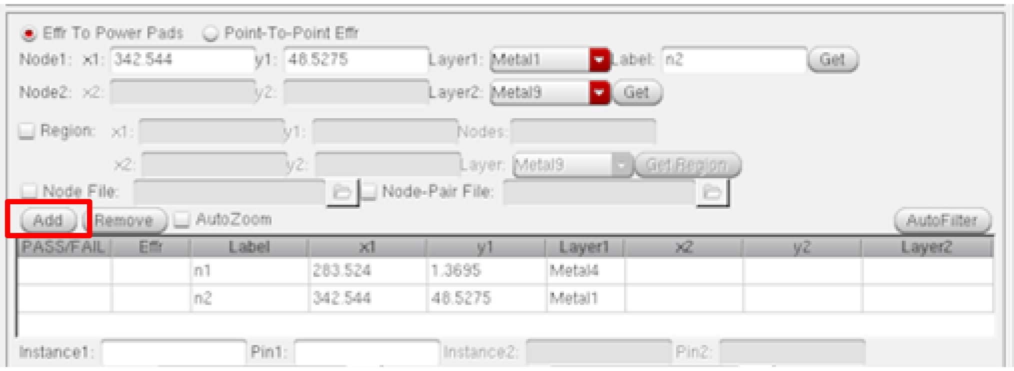
Type	Net	File
XY File	VSS	./design/super_filter_VSS.pp

   d. Set the output directory to: *Reff_VSS_node:*



Output Directory: Reff_VSS_node Output Report Name:

e. Add the node list:



● Effr To Power Pads ○ Point-To-Point Effr

Node1: x1: 342.544 y1: 48.5275 Layer1: Metal1 Label: n2

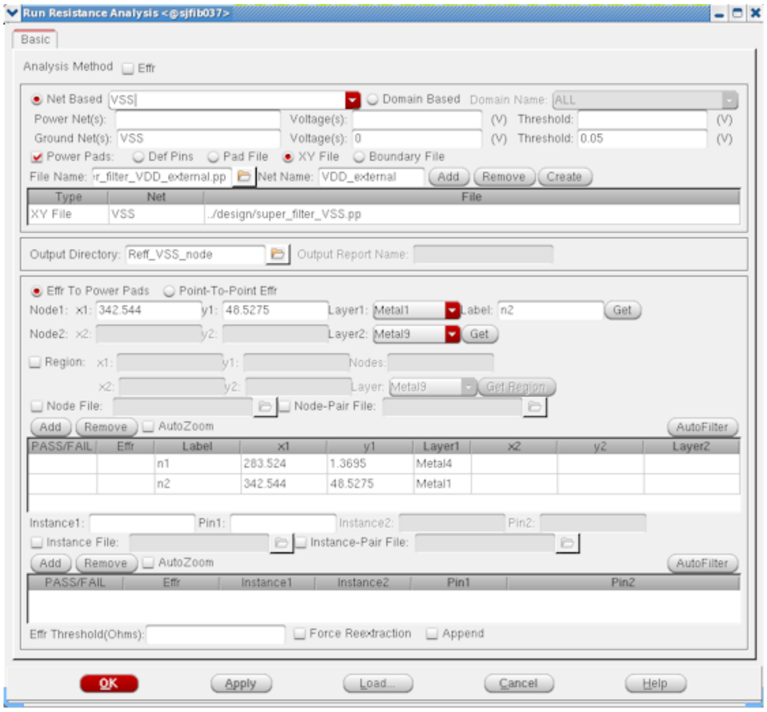
Node2: Layer2: Metal9

Layer: Metal9

PASS/FAIL	Effr	Label	x1	y1	Layer1	x2	y2	Layer2
		n1	283.524	1.3695	Metal4			
		n2	342.544	48.5275	Metal1			

- You can use the *Get* button to get x y coordinates of any node in the design.
- Make sure to select the proper layer name.
- Provide a node name.
- Make sure to use the *Add* button to add the node to the node list.
- You can add as many nodes as possible.

f. Filled form appears as shown in the following image:



Run Resistance Analysis <@sjfib037>

Analysis Method □ Effr

● Net Based VSS ○ Domain Based Domain Name: ALL

Power Net(s): Voltage(s): (V) Threshold: (V)

Ground Net(s): VSS Voltage(s): 0 (V) Threshold: 0.05 (V)

☑ Power Pads: ○ Def Pins ○ Pad File ● XY File ○ Boundary File

File Name: r_filter_VDD_external.pp Net Name: VDD_external

Type	Net	File
XY File	VSS	../design/super_filter_VSS.pp

Output Directory: Reff_VSS_node Output Report Name:

● Effr To Power Pads ○ Point-To-Point Effr

Node1: x1: 342.544 y1: 48.5275 Layer1: Metal1 Label: n2

Node2: x2: y2: Layer2: Metal9

□ Region: Layer: Metal9

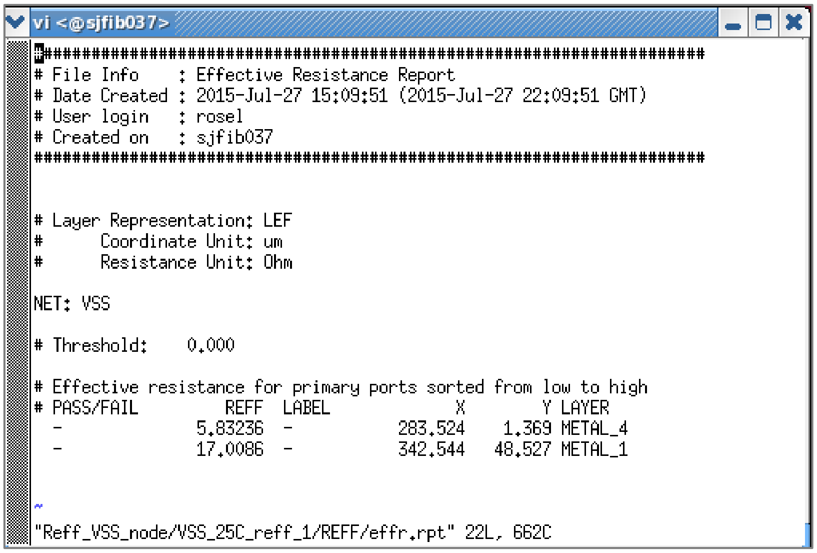
PASS/FAIL	Effr	Label	x1	y1	Layer1	x2	y2	Layer2
		n1	283.524	1.3695	Metal4			
		n2	342.544	48.5275	Metal1			

PASS/FAIL	Effr	Instance1	Instance2	Pin1	Pin2

Effr Threshold(Ohms): □ Force Reextraction □ Append

g. Click *OK* to run.

h. After the job is done, you can view report file: Reff_VSS_node/VSS_25C_reff_1/REFF/effr.rpt

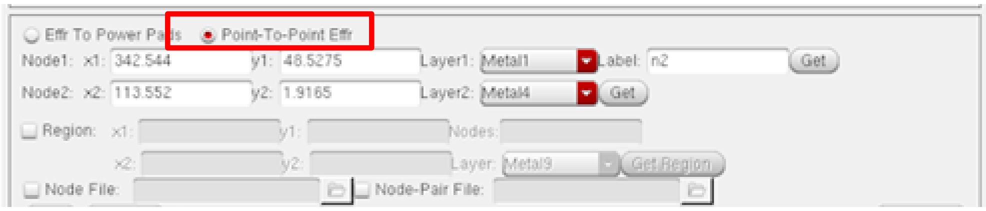


```
vi <@sjfib037>
################################################################################
# File Info     : Effective Resistance Report
# Date Created : 2015-Jul-27 15:09:51 (2015-Jul-27 22:09:51 GMT)
# User login    : rosel
# Created on    : sjfib037
################################################################################


# Layer Representation: LEF
#       Coordinate Unit: um
#       Resistance Unit: Ohm

NET: VSS

# Threshold:     0.000

# Effective resistance for primary ports sorted from low to high
# PASS/FAIL         REFF  LABEL            X          Y LAYER
  -             5.83236  -           283.524     1.369 METAL_4
  -             17.0086  -           342.544    48.527 METAL_1


~
"Reff_VSS_node/VSS_25C_reff_1/REFF/effr.rpt" 22L, 662C
```

i. TCL command:

```
analyze_resistance \
-net              VSS \
-output_dir       Reff_VSS_node \
-node_list        \
      {{ 283.524 1.3695 Metal4 n1 } \
      { 342.544 48.5275 Metal1 n2 }}
```

8. Run Net based node to node effective resistance analysis:
   a. Select the option: *Point-To_Point Effr*:



○ Effr To Power Pads ● Point-To-Point Effr

Node1: x1: 342.544 y1: 48.5275 Layer1: Metal1 Label: n2

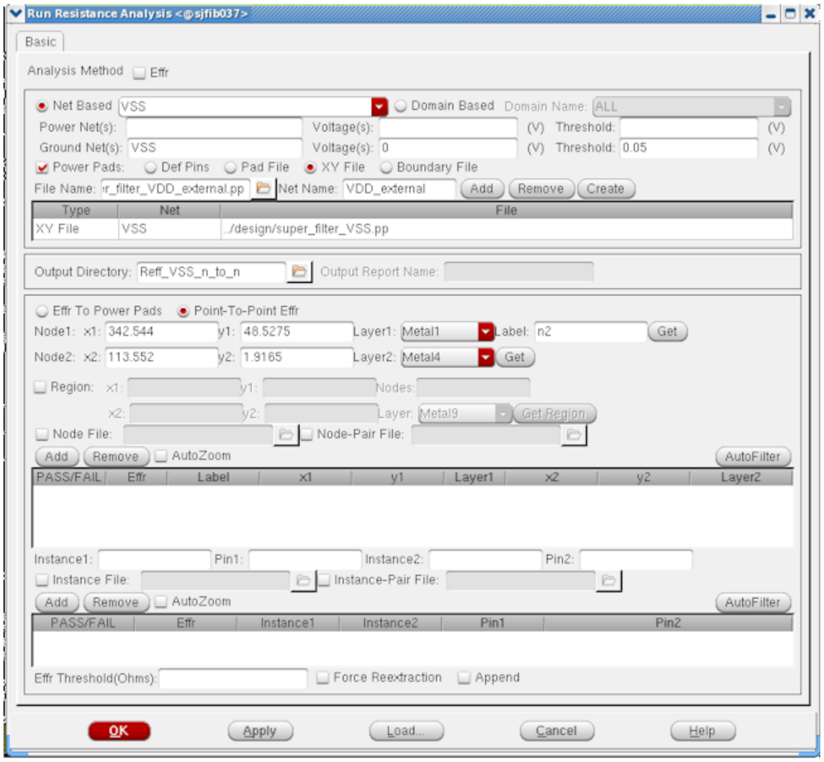
Node2: x2: 113.552 y2: 1.9165 Layer2: Metal4

□ Region: x1: y1: Nodes:

x2: y2: Layer: Metal9

□ Node File: □ Node-Pair File:

   b. Select two points in the design using the *Get* button.
   c. Change the output directory to “ *Reff_VSS_n_to_n*”.
   d. Let the rest of the options as is. The filled form appears as shown in the following image:



Run Resistance Analysis <@sjfib037>

Basic

Analysis Method □ Effr

● Net Based VSS ○ Domain Based Domain Name: ALL

Power Net(s): Voltage(s): (V) Threshold: (V)

Ground Net(s): VSS Voltage(s): 0 (V) Threshold: 0.05 (V)

☑ Power Pads: ○ Def Pins ○ Pad File ● XY File ○ Boundary File

File Name: r_filter_VDD_external.pp Net Name: VDD_external

Type	Net	File
XY File	VSS	../design/super_filter_VSS.pp

Output Directory: Reff_VSS_n_to_n Output Report Name:

○ Effr To Power Pads ● Point-To-Point Effr

Node1: x1: 342.544 y1: 48.5275 Layer1: Metal1 Label: n2

Node2: x2: 113.552 y2: 1.9165 Layer2: Metal4

□ Region: x1: y1: Nodes: x2: y2: Layer: Metal9

□ Node File: □ Node-Pair File:

□ AutoZoom

PASS/FAIL	Effr	Label	x1	y1	Layer1	x2	y2	Layer2

Instance1: Pin1: Instance2: Pin2:

□ Instance File: □ Instance-Pair File:

□ AutoZoom

PASS/FAIL	Effr	Instance1	Instance2	Pin1	Pin2

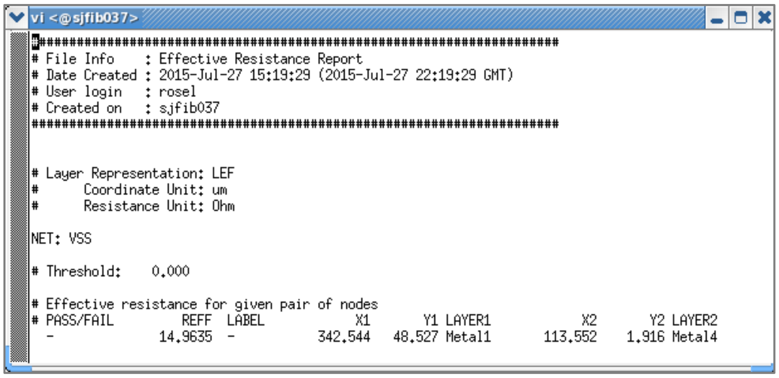
Effr Threshold(Ohms): □ Force Reextraction □ Append

e. Click *OK* to run.

f. TCL command:

```
analyze_resistance \
      -net VSS \
      -output_dir Reff_VSS_n_to_n \
      -node_pair_list \
      {{ 342.544 48.5275 Metal1 113.552 1.9165 Metal4 }}
```

g. View the text report file, "*Reff_VSS_n_to_n/VSS_125C_reff_1/REFF/effr.rpt*", which listed the effective resistance between the two nodes:



vi <@sjfib037>

```
######################################################################
# File Info    : Effective Resistance Report
# Date Created : 2015-Jul-27 15:19:29 (2015-Jul-27 22:19:29 GMT)
# User login   : rosel
# Created on   : sjfib037
######################################################################



# Layer Representation: LEF
#      Coordinate Unit: um
#      Resistance Unit: Ohm

NET: VSS

# Threshold:    0.000

# Effective resistance for given pair of nodes
# PASS/FAIL          REFF  LABEL           X1        Y1 LAYER1          X2        Y2 LAYER2
  -               14.9635  -          342.544    48.527 Metal1     113.552     1.916 Metal4
```

Support

Cadence Online Support provides access to support resources, including an extensive Knowledge Base, access to software updates for Cadence products, and the ability to interact with Cadence Customer Support.

Visit http://support.cadence.com

Feedback

Email comments, questions, and suggestions to content_feedback@cadence.com