Static IR / Power EM Analysis & Debug

RedHawk-SC Modular Training Series

Version : 2020_R3

Ansys

Info For Attendees

- This training session is for 3 hours with 10 mins Q&A session at the end
- To ask questions, please use the Q&A window and one of the panelists will answer it.
- For listening to audio , please use Audio broadcast option

Tutor and Panelists

- **Tutors**
  - Ramesh Agarwal – Lead Product Specialist

- **Panelists**
  - Siddalingesh Tenginakai – Lead Product Specialist
  - Rahul Rajan – Lead Product Specialist
  - Sankar Ramachandran – Director Product Specialist

Prerequisites for the training

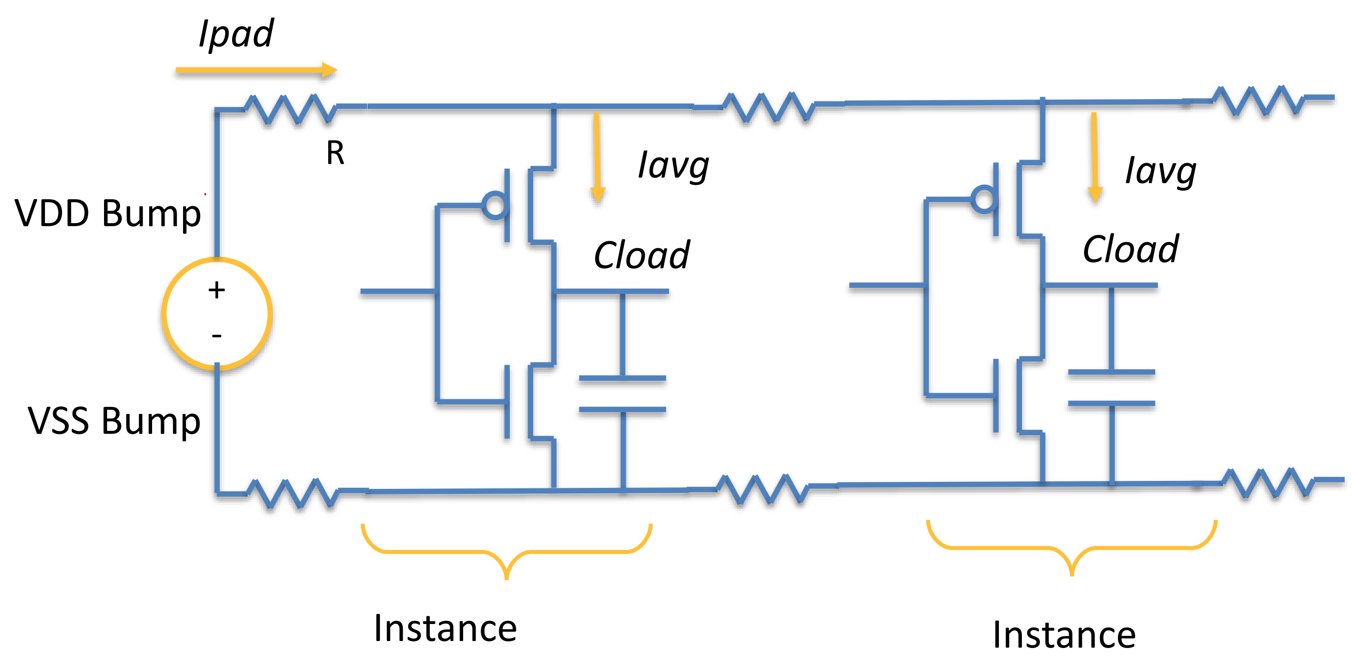
No	Training Program	Expectations – Must Know
1	RedHawk-SC Modular Training • Chapter 01 : Introduction_to_SeaScape	• Reading in input data, performing data integrity checks and creating base views

Training Agenda

- Introduction to Static Analysis
- Getting Familiar with Labs
- Setting the activity in the design
- Power Calculation
- Static Scenario
- Static Voltage Drop Analysis
- Analyzing the results of Static Voltage Drop Analysis
- Basics of Electromigration
- DC Power EM Analysis Flow

Introduction to Static Analysis

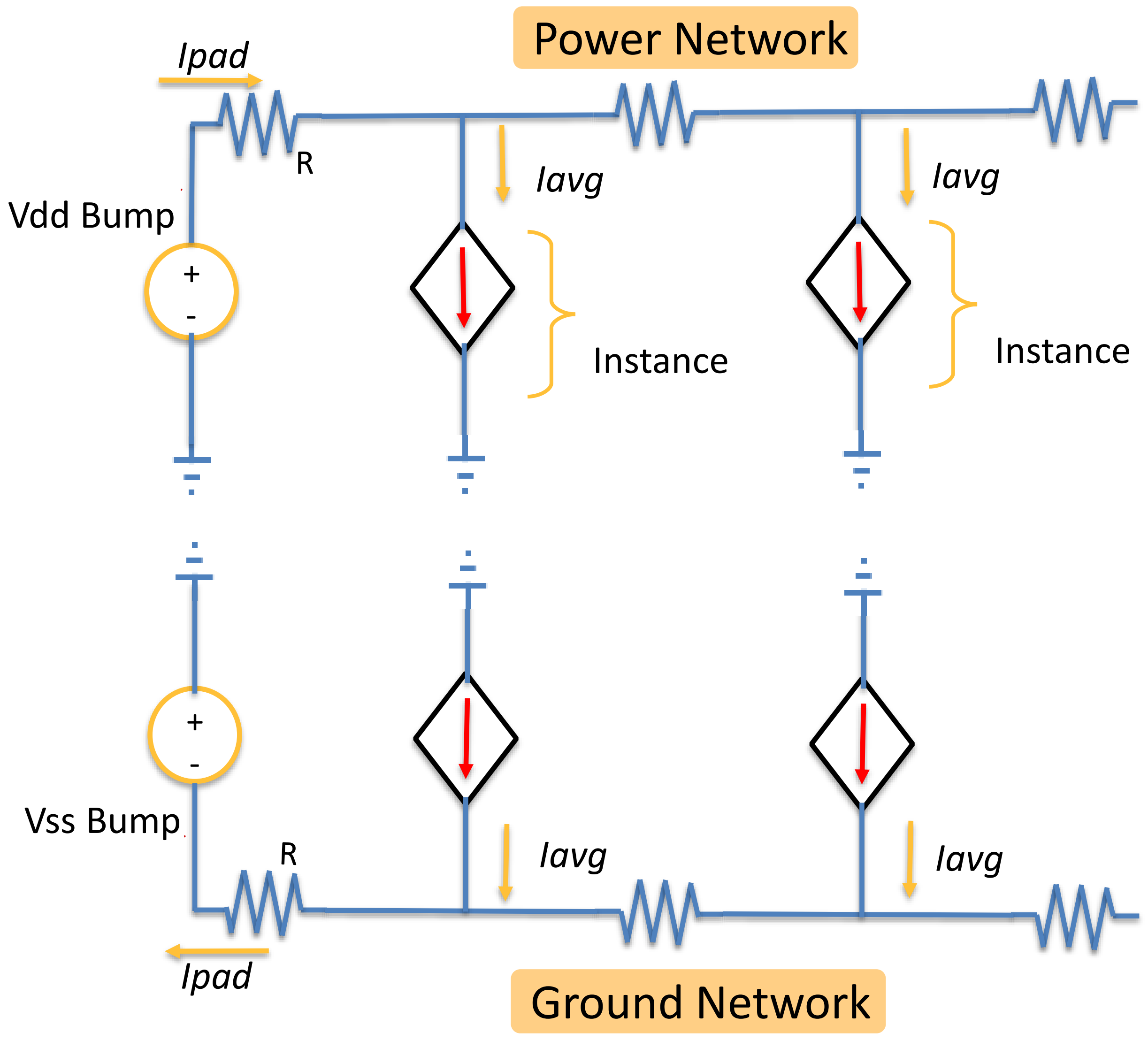
Static Voltage Drop Background



- ➢ On-chip power/ground network ➔ mesh of resistors
- ➢ Instances ➔ DC current sources

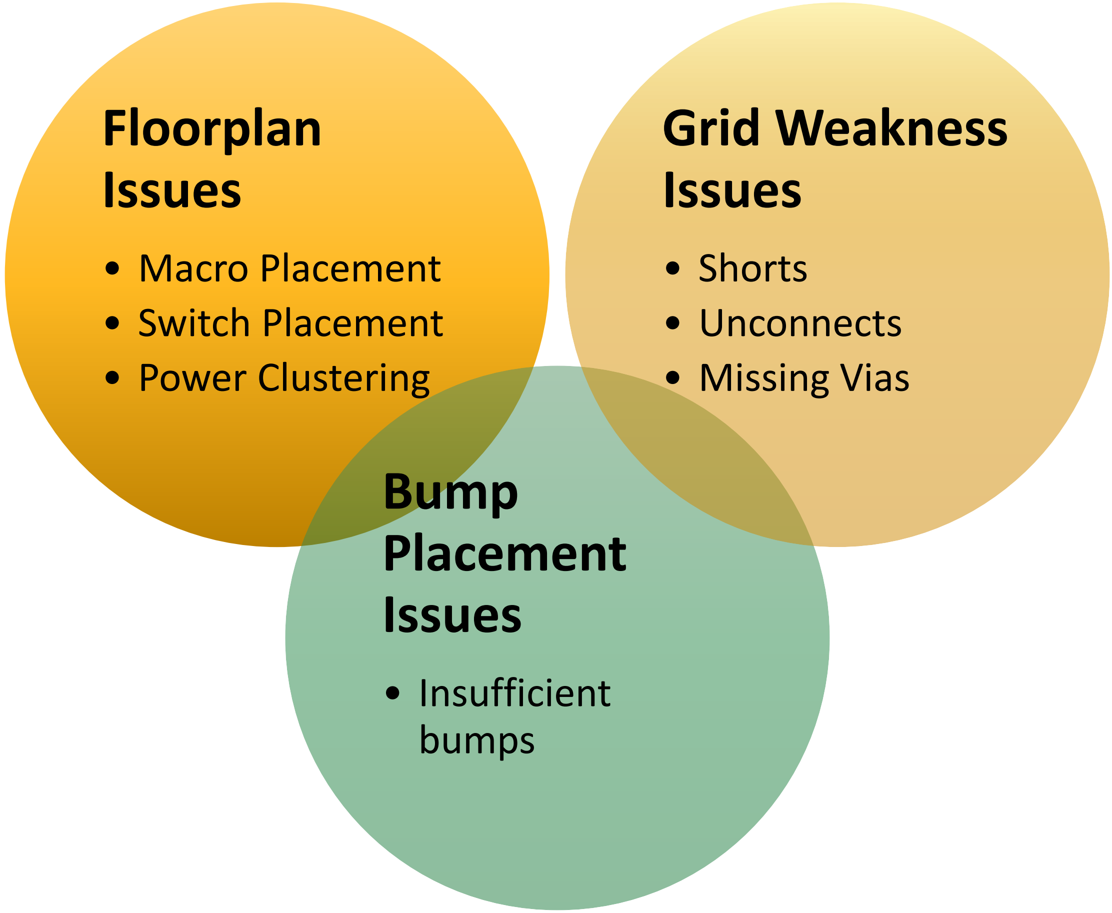
Static Voltage Drop on P/G Network

- Average current is calculated for each instance (using instance average power, $I_{avg} = P_{avg} / V_{supply}$)
- V_{static} is computed at every node (Ohm's law ...)
- Decoupled Circuit is solved
- Each supply net is unique

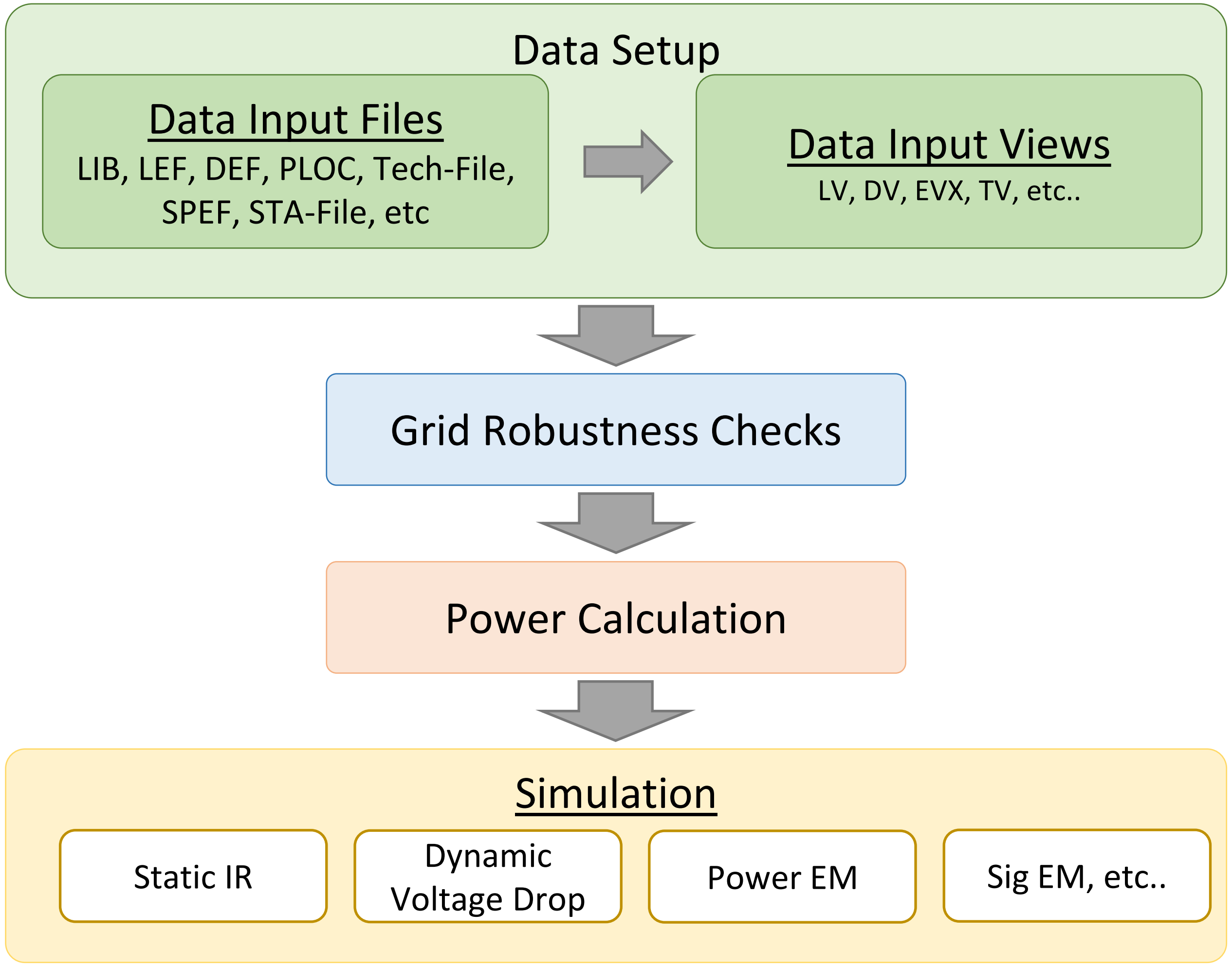


Need for Static Analysis

- Can be done in the initial stage of the design
- Static IR drop analysis is a first-order approximation.
- Identifies Floorplan, Grid Weakness, and Bump Placement Issues in early stage



Typical Power Integrity Analysis Flow



Static Voltage Drop Analysis Flow

Step	Description
SwitchingActivityView	Sets the Toggles on all the instance in design
PowerView	Computes Power Based on Toggle Rates from SwichingActivityView
ScenarioView for Static	Determine Average P/G pin currents from PowerView data
AnalysisView for Static	Static IR drop for each instance PG pin

Getting Familiar with Labs

Getting Familiar with Modular Training Labs

Lab Instructions :

- Download the Galaxy_Training.tar.gz Training Bundle
- Move to Modular_Training/03_Powercalc_And_Static_Analysis

Getting Familiar with Modular Training Labs

Lab Scripts Organization :

Top Wrapper Script

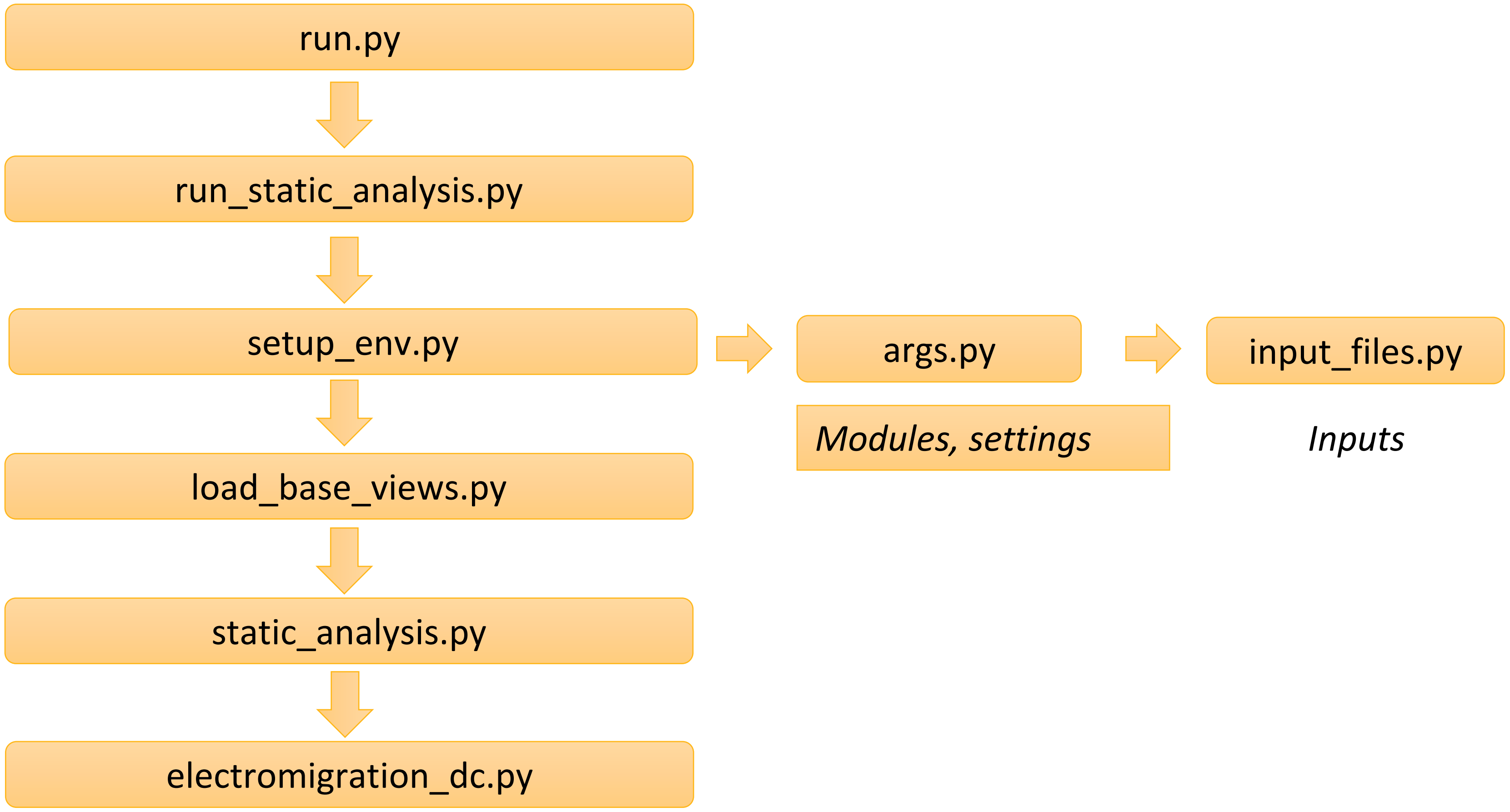
Script that runs all steps

Has modules , settings , input files

loads base views – dv, ev, evx, tv

Create Views like :

- *Switching Activity View*
- *Power View*
- *Scaled Power View*
- *Static Scenario View*
- *Static Analysis View*
- *Power Reports*
- *DC Electro Migration View*



Getting Familiar with Modular Training Lab Scripts

File : run.py

```
include('../scripts/run_static_analysis.py')
```

File : ../scripts/run_static_analysis.py

```
include('setup_env.py')
include('load_base_views.py')
include('static_analysis.py')
include('electromigration_dc.py')
```

- setup_env.py loads worker environment, variables & arguments
- load_base_views.py script loads the base views like dv, ev
- static_analysis.py creates all the required views and reports

Setting up the environment

File : ../scripts/setup_env.py

```
import pprint
open_scheduler_window()

design_data_path = '../../design_data/'

ll = create_local_launcher('local')
register_default_launcher(ll, min_num_workers=10)

include('args.py')
```

Set *design_data_path* variable to central design data path area

Create Launcher for Workers

Set the arguments for various view creation commands in args.py

Launching RedHawk-SC - Get started with your Modular Training

- **Batch mode execution example:**
  - `<path_to_rhsc_installation>/bin/redhawk_sc run.py`
- **Interactive mode execution example:**
  - `<path_to_rhsc_installation>/bin/redhawk_sc -i run.py`
  - It needs an exit() command in script or entered manually to exit the Python shell
- **Connecting to a live RedHawk-SC run:**
  - RedHawk-SC allows querying of data/results from an active session, by remotely attaching to the session
  - Multiple users can attach to the same session from multiple machines for querying/viewing results
  - `<path_to_rhsc_installation>/bin/redhawk_sc -r <gp_dir>`
- **Execution Log Files:**
  - All RHSC log files reside by default in gp<> folder
    - If the run is fired in the same directory, tool will create gp.1, gp.2 incrementally
    - ‘latest.gp’ link will point to the most recent gp directory
  - Main log file for RedHawk-SC would be <gp_directory>/run.log file.

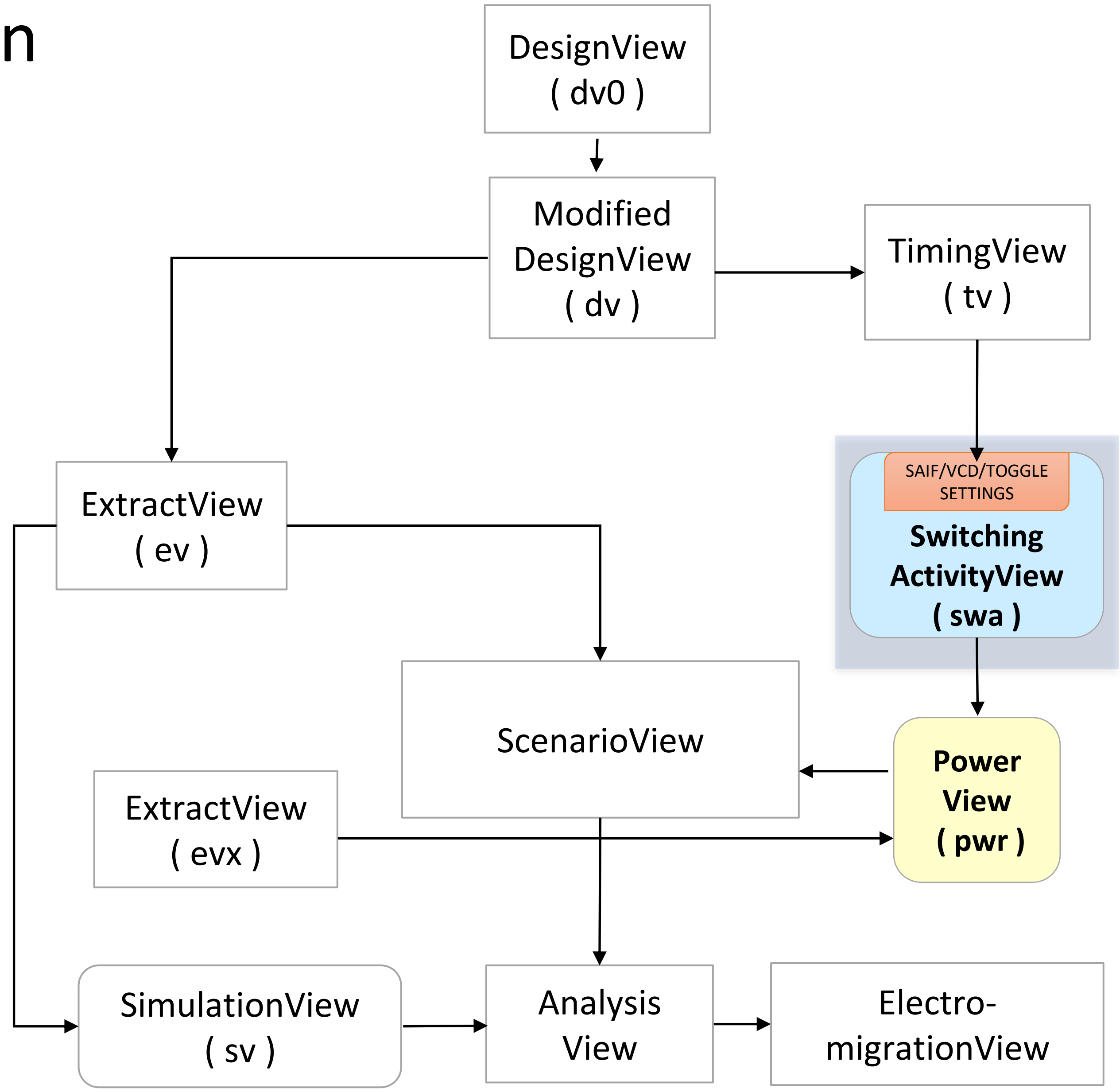
Ansys

Setting the activity in the design

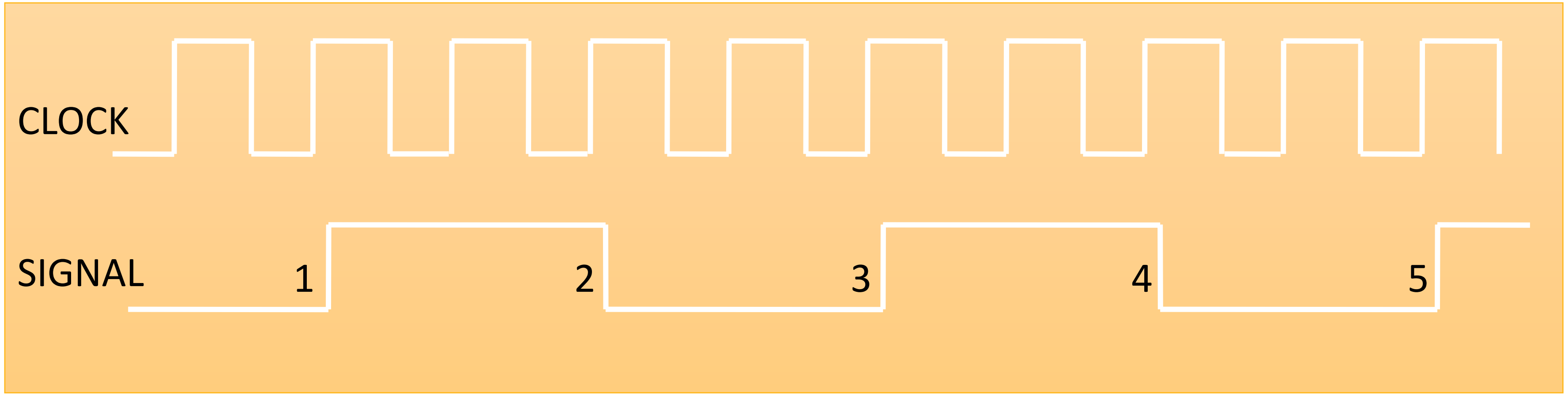
Ansys

Setting the activity in the design

- **SwitchingActivityView**
  - Contains #transitions per second and probabilities of signal remaining high / low
  - Activity can be either
    - Propagated
    - No Propagation / Assign default activity
- **Inputs**
  - DesignView
  - TimingView
  - SAIF (Gate only) [Optional]
  - ValueChangeView [Optional]
  - Gate/RTL VCD/FSDB [Optional]
- **Activity is set through one of the following methods :**
  - User-controlled settings like instance wise toggle rate or global toggle rate
  - SAIF
  - Gate/RTL VCD/FSDB Files or ValueChangeView (VCV)



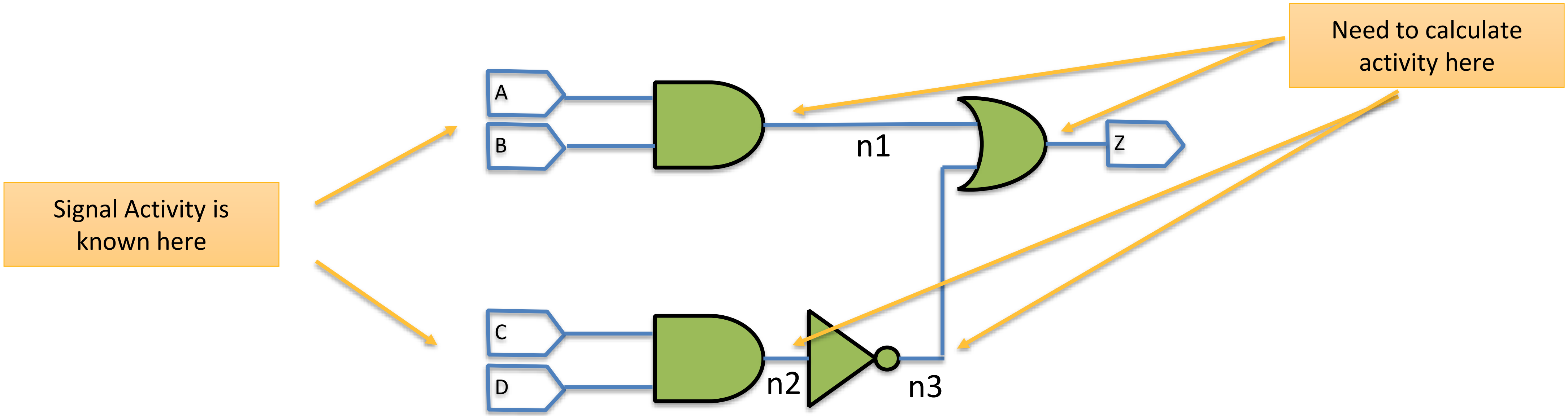
Definition of Toggle Rate



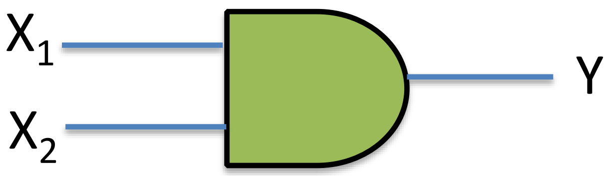
- **A Toggle is 0→1 or 1→0 transition**
- **Toggle rate=(no. of transitions)/(no. of cycles)**
- **Toggle rate CLK=2**
- **Toggle rate SIGNAL=0.5**
- **Toggles per second = no. of toggles in 1 second**

What is Activity Propagation ?

- **Determines the activity at the output of a gate based on the activity at its inputs**
- **Activity is measured in terms of :**
  - One Probability = Fraction of the time signal is high = P(X)
  - Transition Density = Toggles per second == D(X)



Activity Propagation Example



- Let $P(X_i)$ be the fraction of time X_i is high,
- $D(X_i)$ be the toggle rate on X_i

Intuitively :

$P(Y) = P(X_1 \& X_2) = P(X_1) * P(X_2)$

- A transition on X_1 propagates to Y when X_2 is 1 but is blocked when X_2 is 0. Similarly a transition on X_2 propagates to Y only when X_2 is 1.

Therefore

$D(Y) = D(X_1)*P(X_2)+D(X_2)*P(X_1)$

Frequency in STA File

- Clock period and Clock names in design
- Clock Network
- Transition time for Input and Output Pins
- Timing Window for Input and Output Pins
- Constant Signals

CLOCK SECTION

```
### STA Compact Version 1.2
######## Apache Design, Inc. A Subsidiary of ANSYS, Inc. ########
# File Type          : Timing Information (with signal load)
# Design Name        : Galaxy
# Date               : Sun Jan 19 10:37:22 2020
# ATE version        : 19.1.2   RHEL6 (Jun 14 00:53:34 2018)
# pt2timing version : 7.15
##################################################################


# bus_naming_style []
# hierarchy_separator /
# pin_delimiter /

#1) CLOCKS
#CLOCK <rise> <fall> <period> <root> <index>
CLOCK     0.000e-09     4.000e-09     8.000e-09 dco_clk 0
#TIMESCALE 1e-09
#CAPACITANCE_SCALE 1e-15
#RESISTANCE_SCALE 1e+00
```

SLEW/TW SECTION

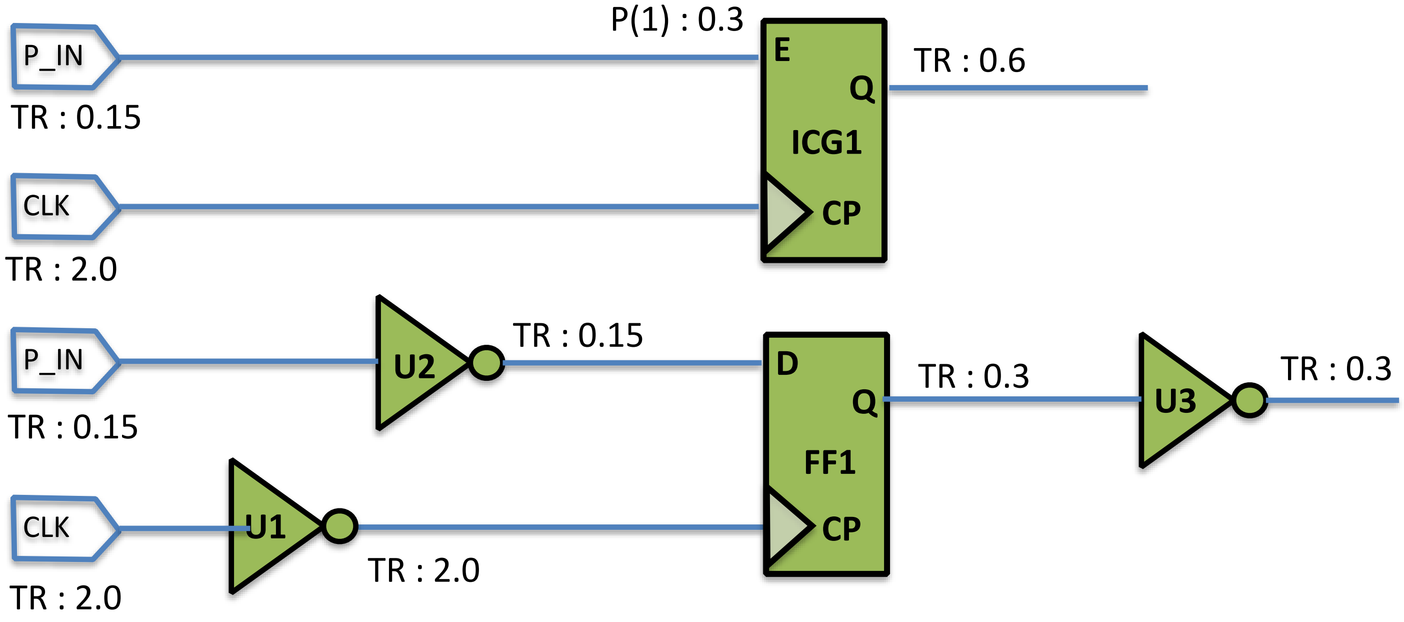
```
#INSTANCE (376829)
#I $<id>/<instance_name>
#C <pin_name> <1|0>
#S <pin_name> <min_r_slew> <max_r_slew> <min_f_slew> <max_f_slew>
#T <pin_name> <is_clock> <min_r> <max_r> <min_f> <max_f> <clk_idx>
I ZBUF_2_inst_65483
S A 0.001 0.001 0.001 0.001
S Z 0.007 0.007 0.006 0.006
T Z 0 4.795 4.804 4.795 4.803 0
```

*For more details refer STA section in Appendix

SwitchingActivityView – propagation_style= “propagate_activity”

- Propagates number of transitions per second
- Propagates probabilities of signals being high and low
- Constants are propagated
- Clock Domain Information is Propagated

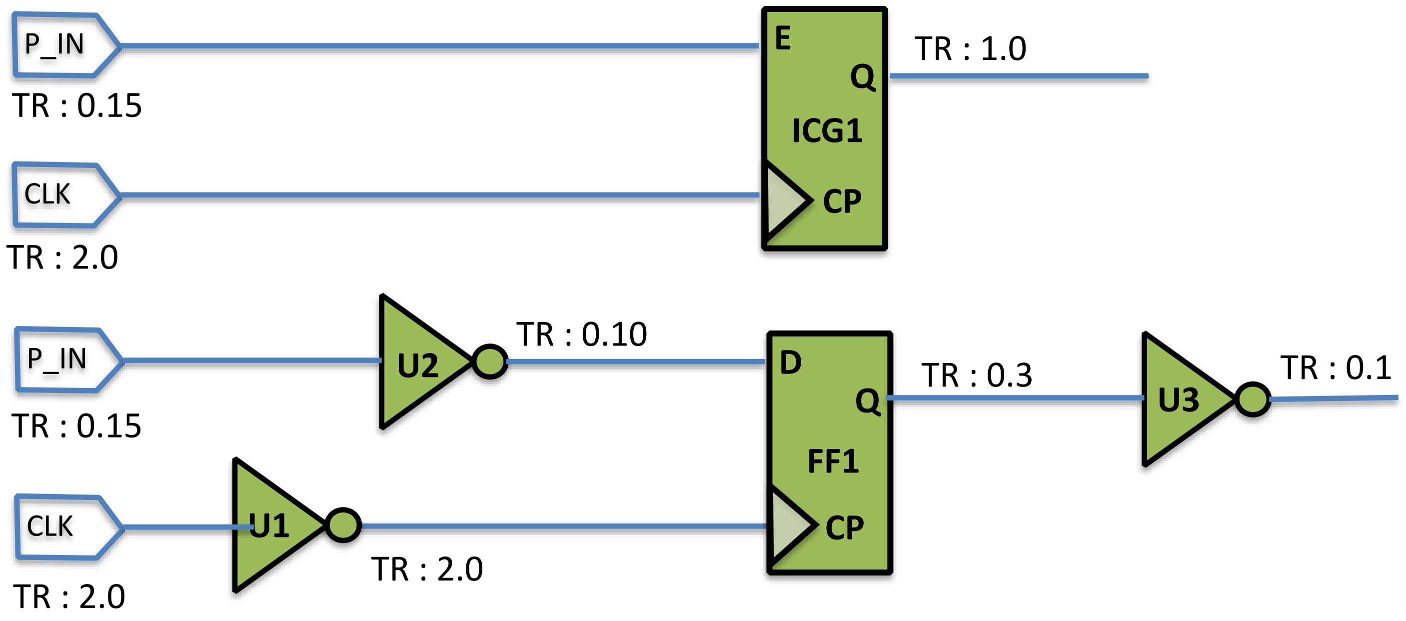
```
swa_object_settings = { 'design_values' : {
'clock_pin_toggle_rate' : 2.0,
'sequential_output_pin_toggle_rate' : 0.3,
'icg_output_pin_toggle_rate' : 1.0,
'icg_enable_pin_one_probability’ : 0.3,
'combinational_pin_toggle_rate' : 0.10,
‘primary_input_port_toggle_rat’ : 0.15,
'propagation_style' : ‘propagate_activity',
}
```



SwitchingActivityView – propagation_style= "assign_default_activity"

- Default activity is assigned
- Constants are propagated
- Clock Domain Information is Propagated

```
swa_object_settings = { 'design_values' : {
'clock_pin_toggle_rate' : 2.0,
'sequential_output_pin_toggle_rate' : 0.3,
'icg_output_pin_toggle_rate' : 1.0,
'icg_enable_pin_one_probability' : 0.3,
'combinational_pin_toggle_rate' : 0.10,
'primary_input_port_toggle_rat' : 0.15,
'propagation_style' : 'assign_default_activity',
}
```



- For propagation_style = "no_propagation" :
  - Clock Domain Information is Not Propagated
  - Huge Performance Benefit
  - SwitchingActivityView can be created with timing view with/without logic_graph = False

Different ways of coming up with Activity

Vectorless

Uniform Toggle Rate	**Vectorless using uniform toggle rate distribution**
Propagate Activity	**Vectorless with activity propagation**

Vector Based

RTL VCD	**Toggles per second is derived from RTL VCD**
Gate VCD	**Toggles per second is derived from Gate VCD**
SAIF	**Toggles per second is derived from SAIF**

Mixed Mode

SAIF – Switching Activity Interchange Format

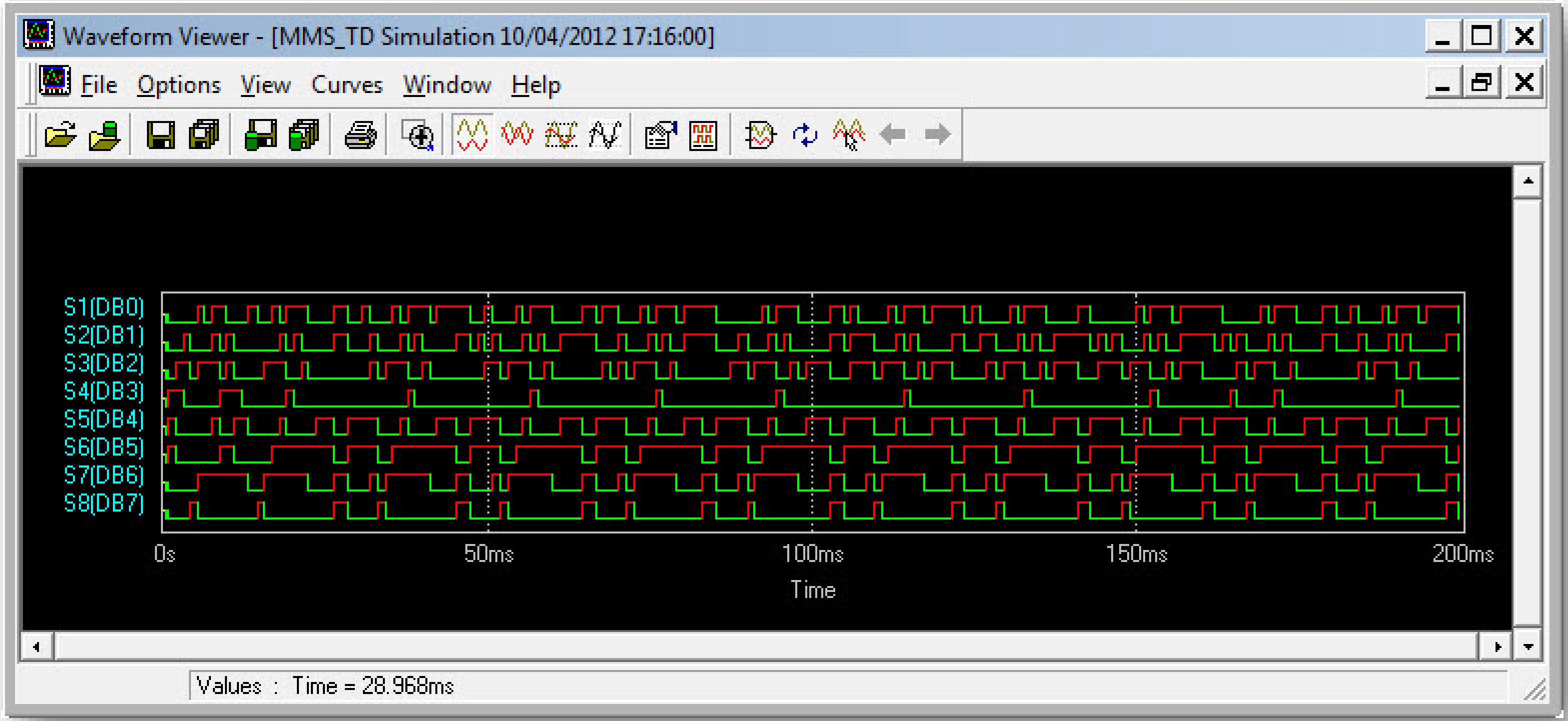
- SAIF contains data captured during simulation for instance pins, including:
- T0: amount of time signal was at logic 0
- T1: amount of time signal was at logic 1
- TC: toggle count over the duration
- Also specifies the duration of the simulation
- SAIF is provided using saif_files arguments which accepts a dict of following keys and values

Argument	Description
file_name	Path to the SAIF file
cell_name	Name of the DEF cell for which to apply the file (optional)
instances	Hierarchical instance object for which to apply the file (optional)
preamble	Name prefix to remove from identifiers in the file (optional)

Example :

```
// SAIF file for seq1_b2
(SAIFILE
  (DATE "Tue Jan 13 15:42:39 PST 2015")
  (SAIFVERSION "2.0")
  (DIRECTION "backward")
  (TIMESCALE 1 ns)
  (DURATION 100)
  (INSTANCE U3
    (INSTANCE FF1
      (NET
        (Q (T0 1) (T1 99) (TC 200))
      )
    )
    (INSTANCE FF2
      (NET
        (Q (T0 2) (T1 98) (TC 200))
      )
    )
  )
  (INSTANCE U4
    (INSTANCE FF1
      (NET
        (Q (T0 3) (T1 97) (TC 200))
      )
    )
    (INSTANCE FF2
      (NET
        (Q (T0 4) (T1 96) (TC 200))
      )
    )
  )
)
```

VCD / FSDB file



VCD stands for "Value Change Dump"
VCD File contains the Waveform for different nets and inst/pins in the design
FSDB (Fast signal database) contains same information, but in binary format
FSDB file size is much smaller than VCD

VCD / FSDB file

- VCD / FSDB file is passed to SwitchingActivityView using vcd_files argument which accepts a dict of following keys and values
- Name mapping may be required for RTL VCD / FSDB
- SwitchingActivityView count the number of toggles from vcd
- SwitchingActivityView also counts the following
    - Toggles Per Second
    - One Probability

Key	Description
file_name	Path to the VCD file
instance_name	Hierarchical instance name for which to apply the file (optional)
preamble	Name prefix to remove from identifiers in the file (optional)
start_time	Starting time value to consider in VCD
stop_time	Ending time value to consider in VCD
top_only	Only read signals at top level of hierarchy (optional)

VCD / FSDB file

- VCD / FSDB file can be passed via value_change_view data as well
- More details on creating Value_change_view is discussed in Dynamic Training

Key	Description
'view'	ValueChangeView tag name, whose data we want to apply
'slice_name'	Slice of the specified ValueChangeView
'time_delay'	Amount of time to shift events/waveforms. Optional.
'time_scale_factor'	All time values for signal changes will be multiplied by this factor. Optional.

Loading Base Views

File : ../scripts/load_base_views.py

```
db = open_db('../db')
# Auto-load view tags from an existing db
# Needed only for incremental(jump-start) runs
populate_view_tags()
```

- Auto-load view tags from an existing db with base views
- Auto-loading of view tags needed for only incremental jump-start runs

Create Switching Activity View

File : scripts/static_analysis.py

```
swa_mixed = db.create_switching_activity_view(timing_view=tv,**swa_mixed_args)
```

Argument	Description
timing_view	TimingView object (type=TimingView, required=True)
saif_files	List of SAIF files (type=object, default_value=None, constraint="list")
vcd_files	List of VCD files (type=list, default_value=None, constraint="VCD files")
value_change_data	List of dicts specifying ValueChangeView slices (type=list, default_value=None)

Ansys

Switching Activity View Arguments

File : scripts/args.py

```
swa_mixed_args = dict(
    vcd_files = vcd_files_swa,
    object_settings = swa_object_settings,
    tag='swa_mixed',
    options=options)
```

```
swa_object_settings = { 'design_values' : {
'clock_pin_toggle_rate' : 2.0,
'sequential_output_pin_toggle_rate' : 0.15,
'macro_output_pin_toggle_rate' : 0.15,
'icg_output_pin_toggle_rate' : 1.0,
'combinational_pin_toggle_rate' : 0.15,
'propagation_style' : 'no_propagation',
'clock_period' : 8e-09,        }}
```

Toggles can be specified in various scope :

- Design Level
- Block Specific
- Cell Specific
- Leaf Instance wise

For more details , see : **help(SeaScapeDB.create_switching_activity_view)**

Viewing Switching Activity View attributes

```
swa_mixed.get_attributes(Instance("core0.regfile_pro
gram_memory.MUX2_X1_1377"))
{Pin('A'): {'clock_name': 'dco_clk',
          'clock_period': 7.9999997737744548e-09,
          'one_probability': 0.5,
          'source': 'default',
          'toggles_per_second': 18750002.0,
          'zero_probability': 0.5},

Pin('Z'): {'clock_name': 'dco_clk',
          'clock_period': 7.9999997737744548e-09,
          'one_probability': 0.5,
          'source': 'default',
          'toggles_per_second': 18750002.0,
          'zero_probability': 0.5}}
```

```
swa_mixed.get_attributes(Instance("core3/regfile_pro
gram_memory.MUX2_X1_3464"))
{Pin('Z'): {'clock_name': 'dco_clk',
          'clock_period': 7.9999997737744548e-09,
          'one_probability': 0.570888876914978,
          'source': 'VCD/SAIF',
          'toggles_per_second': 67500000.0,
          'zero_probability': 0.429111123085022}}
```

VCD activity is annotated for core3

This shows that source of activity is default_toggle_rate *

Toggle Rate = toggles_per_second * clock_period = 0.15

*Refer to Appendix for sources in SwitchingActivityView

Power Calculation

Power Calculation Basics

Leakage Power + Internal Power + Switching Power = Average Power

```
leakage_power ()  {
  value : 0.000000E+00;
  when : "A&!B";
  power_level : "VDDNW";
}

leakage_power ()  {
  value : 4.430000E+02;
  when : "A&!B";
  power_level : "VDD";
}
```

From .lib

```
internal_power ()  {
  power_level : "VDDNW";

  rise_power (inputpower_trans4)  {
    index_1 ("25.0000,50.0000,625.0000,1250.0000");
    values ("0.0040,  0.0040,  0.0040,  0.0040");
  }

  fall_power (inputpower_trans4)  {
    index_1 ("25.0000,50.0000,625.0000,1250.0000");
    values ("0.0040,  0.0040,  0.0040,  0.0040");
  }
}
```

From .lib

Switching Power = ½ * Cap * V^2 * Freq * Toggle_Rate

From SPEF

From Timing File

Internal Power = Internal Energy(in .lib) * Freq * Toggle Rate

P(total) = P(Leakage) + ½ * (Internal-energy * Freq * TR) + ½ * C * V^2 * Freq * TR

Leakage Power Calculation

- LIB File has leakage defined for various states and average of all states
- AND Gate Example

```
cell (AND2_X1) {
  drive_strength      : 1;
  area                : 1.064000;
  pg_pin(VDD) {
      voltage_name : VDD;
      pg_type      : primary_power;
  }
  pg_pin(VSS) {
      voltage_name : VSS;
      pg_type      : primary_ground;
  }
  cell_leakage_power  : 25.066064;
  leakage_power () {
      when           : "!A1 & !A2";
      value          : 20.324370;
  }
  leakage_power () {
      when           : "!A1 & A2";
      value          : 30.850688;
  }
  leakage_power () {
      when           : "A1 & !A2";
      value          : 20.622958;
  }
  leakage_power () {
      when           : "A1 & A2";
      value          : 28.466240;
  }
  pin (A1) {
      direction       : input;
      related_power_pin      : "VDD";
      related_ground_pin     : "VSS";
```

Average of all states

Leakage for A1=0 A2=0

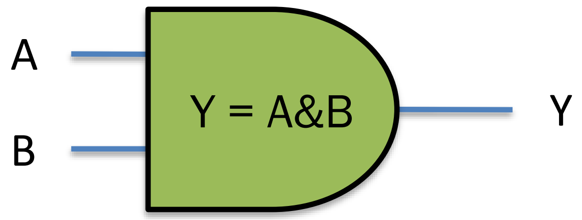
Leakage for A1=0 A2=1

Leakage for A1=1 A2=0

Leakage for A1=1 A2=1

Internal Power Calculation

- LIB File has Internal Energy characterized for various events
- Internal Energy is characterized as a function of load and transition time
- AND Gate Example



A	B	Y	ARC Name
RISE	0	0	A RISE (Y=0)
FALL	0	0	A FALL (Y=0)
RISE	1	RISE	A->Y RISE
FALL	1	FALL	A->Y FALL
0	RISE	0	B RISE (Y=0)
0	FALL	0	B FALL (Y=0)
1	RISE	RISE	B->Y RISE
1	FALL	FALL	B->Y FALL

*For more details refer to section of Internal Power Calculation from Liberty file in Appendix

Instance Power File (IPF)

- IPF file can be provided as direct power information to RedHawk-SC.
- IPF files are written by 3rd Party Power Calculation Tools.
- IPF file can be written from PowerView as well.
- Support for 9-Column IPF file (detailed) and 3-Column IPF (total power per pin) files

3 column IPF file

```
# instance_name total_power pin_name
ZBUF_314_inst_65462 4.67768529688e-07 VDD
ZBUF_9013_inst_51856 1.68285282598e-06 VDD
inv_drc_cln62326 5.78630315431e-05 VDD
ZBUF_8403_inst_51899 5.38082758794e-07 VDD
ZBUF_3260_inst_51931 1.10659289021e-06 VDD
ZBUF_537_inst_51950 5.44689214621e-07 VDD
cts_inv_583961390 8.77659767866e-05 VDD
ZBUF_2517_inst_51972 1.58077830292e-06 VDD
ZBUF_2644_inst_51974 6.36480422145e-07 VDD
cts_inv_584361394 8.77179190866e-05 VDD
cts_inv_584461395 7.40412360756e-05 VDD
cts_inv_584961400 7.24101919332e-05 VDD
```

9 column detailed IPF file

```
# instance_name pin_name voltage toggle_rate frequency total_power switching_power internal_power leakage_power
ZBUF_314_inst_65462 VDD 1.10000002384 0.15000000596 125000000.0 4.67768529688e-07 3.38084106488e-07 9.91261970285e-08 3.05582155136e-08
ZBUF_3658_inst_51786 VDD 1.10000002384 0.15000000596 125000000.0 1.3510846486e-07 8.57065742821e-08 3.81877960365e-08 1.12140927655e-08
ZBUF_5852_inst_51787 VDD 1.10000002384 0.15000000596 125000000.0 2.21814434553e-06 1.9550336674e-06 1.76988777412e-07 8.6121801246e-08
ZBUF_9013_inst_51856 VDD 1.10000002384 0.15000000596 125000000.0 1.68285282598e-06 1.54588008172e-06 1.06414510981e-07 3.05582155136e-08
inv_drc_cln62326 VDD 1.10000002384 2.0 125000000.0 5.78630315431e-05 5.57769817533e-05 1.97122358259e-06 1.1482629958e-07
ZBUF_8403_inst_51899 VDD 1.10000002384 0.15000000596 125000000.0 5.38082758794e-07 4.73716312399e-07 5.31523305369e-08 1.12140927655e-08
ZBUF_3260_inst_51931 VDD 1.10000002384 0.15000000596 125000000.0 1.10659289021e-06 9.91275101114e-07 9.24000715941e-08 2.29176180255e-08
ZBUF_537_inst_51950 VDD 1.10000002384 0.15000000596 125000000.0 5.44689214621e-07 4.88130126541e-07 4.53450006432e-08 1.12140927655e-08
cts_inv_583961390 VDD 1.10000002384 2.0 125000000.0 8.77659767866e-05 8.34217280499e-05 4.1145999603e-06 2.2965144808e-07
ZBUF_2517_inst_51972 VDD 1.10000002384 0.15000000596 125000000.0 1.58077830292e-06 1.45164528931e-06 9.85747590221e-08 3.05582155136e-08
ZBUF_2644_inst_51974 VDD 1.10000002384 0.15000000596 125000000.0 6.36480422145e-07 5.86834232763e-07 3.84321197089e-08 1.12140927655e-08
cts_inv_584361394 VDD 1.10000002384 2.0 125000000.0 8.77179190866e-05 8.33734375192e-05 4.11482778873e-06 2.2965144808e-07
cts_inv_584461395 VDD 1.10000002384 2.0 125000000.0 7.40412360756e-05 6.95760536473e-05 4.23553137807e-06 2.2965144808e-07
cts_inv_584961400 VDD 1.10000002384 2.0 125000000.0 7.24101919332e-05 7.03441619407e-05 1.95119832824e-06 1.1482629958e-07
```

Mode Control

- The current for a macro is highly dependent on its mode
- Mode Control is used in PowerView to specify the switching states i.e mode of macros.
- Mode Sequence or Mode Probabilities can be set for a given scope (design, block, cell , instance level)
- Implicit modes available are :

Mode name	Description
`_low_energy_mode_`	Lowest Energy Mode
`_median_energy_mode_`	Median Energy Mode
`_high_energy_mode_`	Highest Energy Mode
`_leakage_only_mode_`	Only Leakage is Consumed
`_off_mode_`	Off Mode , i.e not even Leakage

- Explicit modes from APL and Mode definitions is also supported
- Power View will compute the average power of given mode sequence / mode probabilities and assign this power to Macro
- Instances using Mode control will have 'ModeControl' as source in power attributes.

Mode Sequence and Mode Probabilities

Mode Sequence:

Read	Read	Write	Standby	Read	Read	Write	Standby	…

Mode Probabilities

Read	0.2
Write	0.3
Standby	0.5

Mode for an instance will be chosen randomly per active clock edge, based on defined probabilities

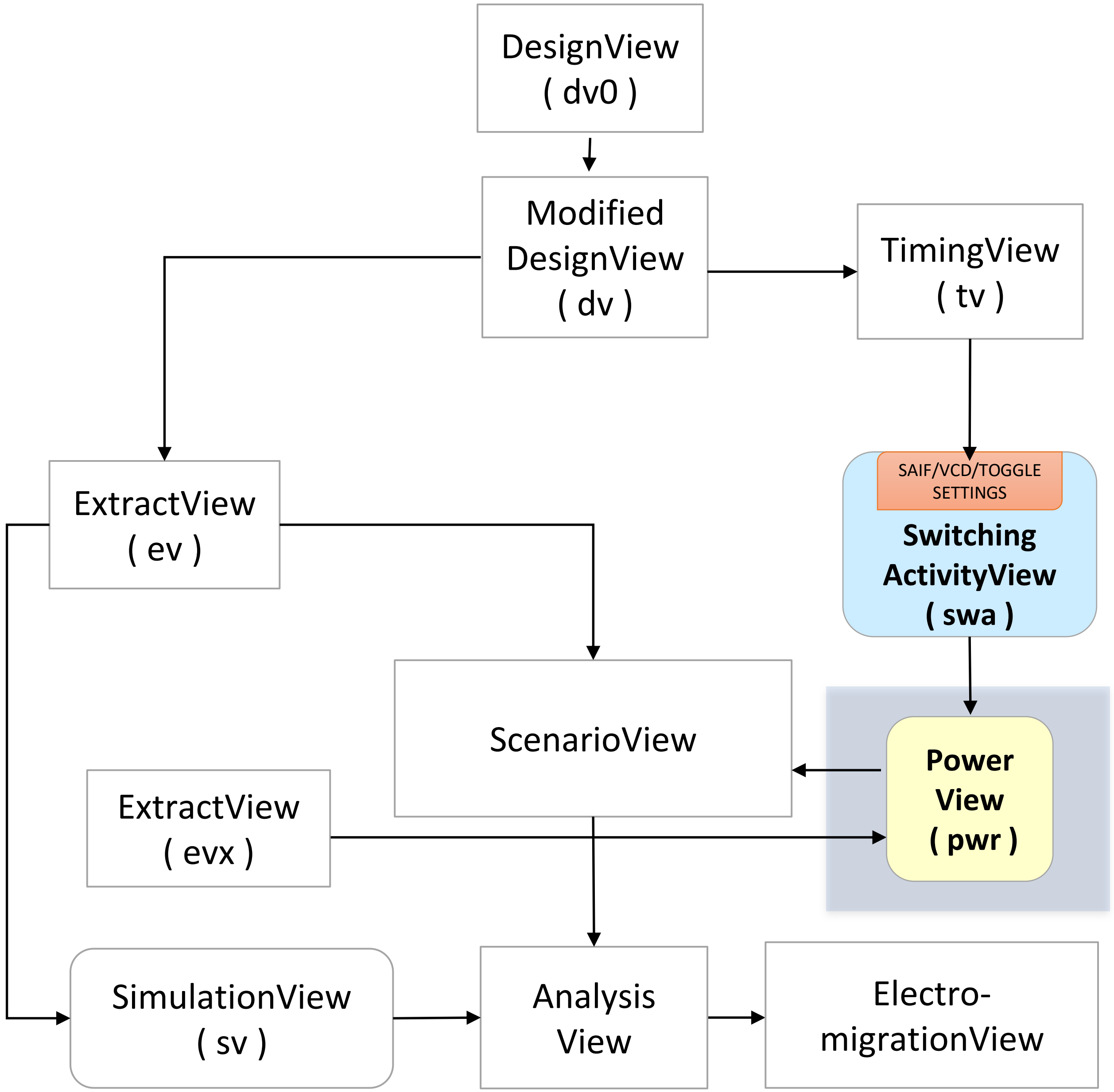
Mode Control . . . continued

- Example :

```
object_settings = {
'leaf_instance_values' : [
{'instances' : Instance('core1.program_memory'), 'mode_control' : { 'mode_sequence' : ['MEM_READ'],  }},
{'instances' : Instance('core2.data_memory'),    'mode_control' : { 'mode_sequence' : ['MEM_WRITE'], }},
{'instances' : Instance('core3/data_memory'),    'mode_control' : { 'mode_sequence' : ['MEM_WRITE'], }},
{'instances' : Instance('core1.data_memory'),    'mode_control' : { 'mode_sequence' : ['_high_energy_mode_'],}},
{'instances' : Instance('core0.data_memory'),    'mode_control' : { 'mode_sequence' : ['_leakage_only_mode_'], }},
{'instances' : Instance('core2.program_memory'), 'mode_control' : { 'mode_sequence' : ['_low_energy_mode_'], }},
{'instances' : Instance('core3/program_memory'), 'mode_control' : { 'mode_sequence' : ['_median_energy_mode_'] }},
{'instances' : Instance('core0.program_memory'), 'mode_control' : { 'mode_sequence' : ['_high_energy_mode_'] }},]}
```

Performing Power Calculation

- **PowerView**
  - Contains the power per instance and its related info like load, transition time, toggle rate, frequency, voltage levels, etc
- **Inputs for creating PowerView:**
  - DesignView
  - SwitchingActivityView (Toggles)
  - Instance Power File, IPF (Direct power information)
  - ExtractView (SPEF based or RedHawk-SC native for load information)
  - TimingView (Timing)
  - Settings for macro power control through mode control
  - For more details , see : **help(SeaScapeDB.create_power_view)**



Performing Power Calculation

File : scripts/args.py

```
pwr_mixed_args = dict(
    voltage_levels=voltage_levels,
    object_settings = object_settings,
    tag='pwr_mixed',
    options=options)
```

File : scripts/static_analysis.py

```
pwr_mixed = db.create_power_view(switching_activity_view=swa_mixed, external_parasitics=evx, **pwr_mixed_args)

reports_dir = 'reports/powercalc'

gp_util.makedir(reports_dir)

emir_reports.write_instance_power_report_and_summary(pwr_mixed, reports_dir+'/instance_power.rpt',
reports_dir+'/power_summary.rpt')
```

Viewing PowerView attributes

```
pwr_mixed.get_attributes(Instance("core3/program_memory"))
{Pin('vdd'): {'clock_pin_power': 0.00016499549383297563,
            'frequency': 125000000.0,
            'internal_power': 0.0001634641521377489,
            'leakage_power': 8.49892785481643e-06,
            'source': 'ModeControl',
            'switching_power': 1.5313393078031368e-06,
            'toggle_rate': 0.60999995470047,
            'total_power': 0.00017349442350678146,
            'voltage': 1.100000023841858}}
```

Power is computed for macros-based mode control provided in Power View via Object Settings*

```
pwr_mixed.get_attributes(Instance("core3/regfile_program_memory.MUX2_X1_3464"))
{Pin('VDD'): {'clock_pin_power': 0.0,
            'frequency': 125000000.0,
            'internal_power': 1.9928337735564128e-07,
            'leakage_power': 3.856697716742019e-08,
            'source': 'SwitchingActivity',
            'switching_power': 9.795139988000301e-08,
            'toggle_rate': 0.5399999618530273,
            'total_power': 3.3580175795577816e-07,
            'voltage': 1.100000023841858}}
```

This shows that Power is computed based on toggle rate from Switching Activity View*

* For more details , refer Appendix Sources in PowerView

Viewing PowerView results

```
pwr_mixed.get_total_power()
{Net('core3/VDD_INT'): [{'leakage_power': 0.0035269377985969186,
                         'internal_power': 0.027774976566433907,
                         'total_power': 0.08402520697563887,
                         'switching_power': 0.0527232950553298}],
 Net('VDD'):            [{'leakage_power': 0.011251124528294909,
                         'internal_power': 0.045948175243328339,
                         'total_power': 0.12869410832713513,
                         'switching_power': 0.071494808450430 55}]}
```

Get total Power per Net

```
pwr_mixed.get_output_total_capacitance(Instance("_56833_"))
{Pin('Q'): 4.048312317425813e-14,
 Pin('QN'): 0.0
}
```

Get each output pin's total capacitance as a dict of {Pin : capacitance}

Ansys

Looking At Power Calculation Reports

Command Used :

```
emir_reports.write_instance_power_report_and_summary(pwr_mixed, reports_dir+'/instance_power_mixed.rpt', reports_dir+'/power_summary_mixed.rpt’)
```

File : power_summary_mixed.rpt

```
**** RedHawk-SC Power Summary Report ****
Created: Sun Sep 20 21:18:41 2020

A total of 1185384 instances were summarized for this report while 0 were omitted due to missing power data (100.00% coverage).

A total of 0 pins were omitted because they were not attached to a power domain.

***

grouping              clock_pin_power(W)  internal_power(W)  leakage_power(W)  switching_power(W)  total_power(W)  percent_power(%)  pin_count  instance_count

*** Power Domains:
VDD                        0.025298            0.045948          0.011256          0.071495          0.1287          60.50        912472      909926
core3/VDD_INT              0.0078372           0.027775          0.0035269         0.052723          0.084025        39.50        275458      275458
Total                      0.033135            0.073723          0.014782          0.12422           0.21272        100.00        1187930     1185384

*** Frequency Domains:
1.25e+08                   0.033135            0.073723          0.013739          0.12422           0.21168         99.51        377445      377445
0                          0                   7.9383e-08        0.0010435         0                 0.0010436        0.49        810485      807939
Total                      0.033135            0.073723          0.014782          0.12422           0.21272        100.00        1187930     1185384

*** User Defined Groups:
combinational logic        0                   0.021952          0.0090181         0.11249           0.14346         67.44        310766      308220
sequential logic           0.032636            0.051279          0.0057205         0.011719          0.068719        32.30        71800       71800
memory                     0.00049972          0.00049296        4.3939e-05        6.9543e-06        0.00054366       0.26        8           8
decap/filler               0                   0                 0                 0                 0                0.00        805356      805356
Total                      0.033135            0.073723          0.014782          0.12422           0.21272        100.00        1187930     1185384

**** End Report ****
```

*For more details, refer to Appendix CellType Definition in Power Calculation Reports Section

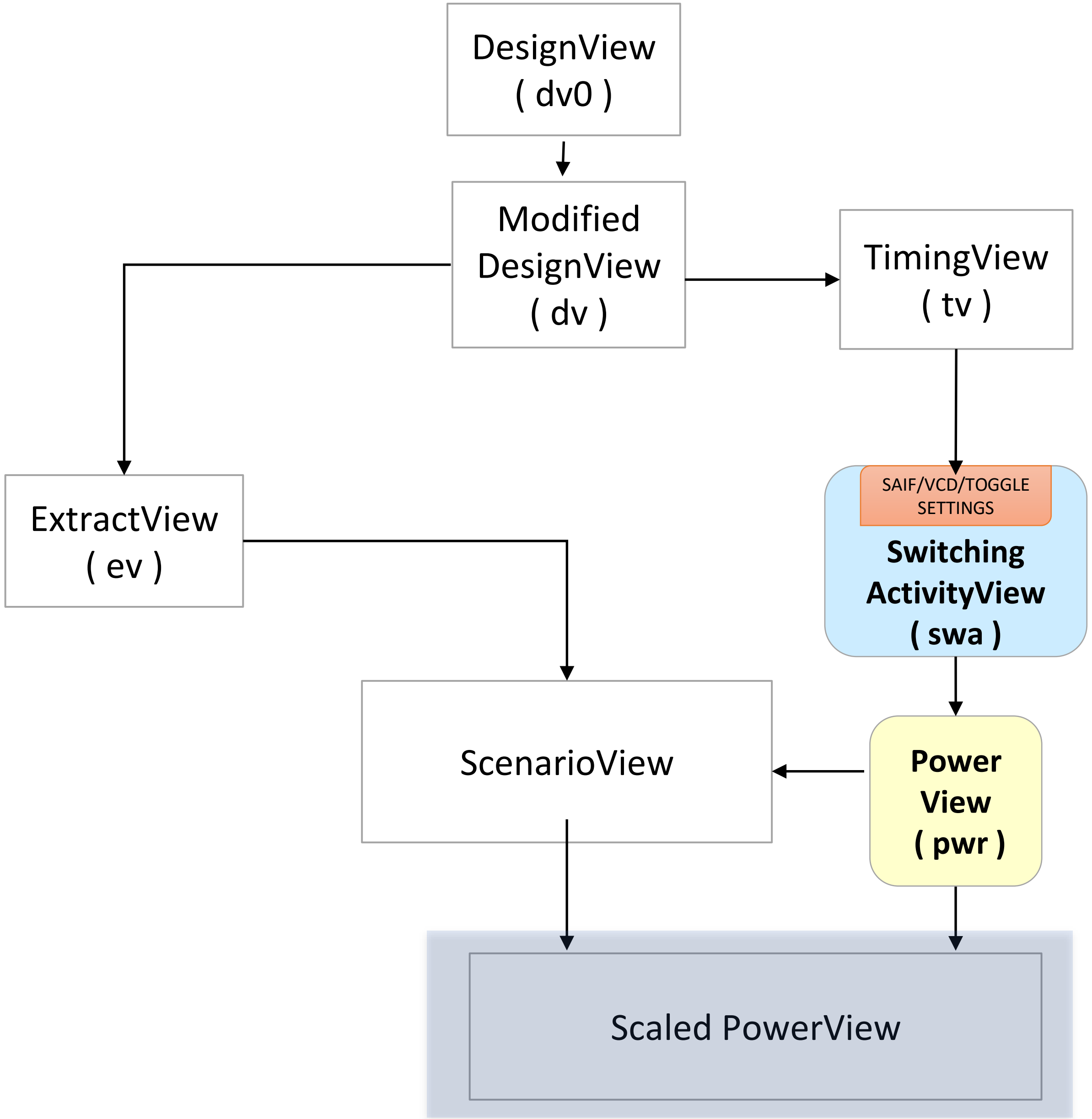
Looking At Power Calculation Reports

File : instance_power_mixed.rpt

```
# clock_pin_power is included in internal_power
# pin     domain   frequency  toggle_rate  clock_pin_power  internal_power  leakage_power  switching_power  total_power  voltage  cell_name   instance
#                    (Hz)                        (W)              (W)            (W)             (W)            (W)         (V)
  VDD      VDD     1.25e+08      2.00              0          1.012e-05      4.593e-07       0.0001029     0.0001135     1.10    INV_X32    cts_inv_590761458
  VDD      VDD     1.25e+08      2.00              0          8.234e-06      4.593e-07       0.0001028     0.0001115     1.10    INV_X32    cts_inv_528960840
  VDD      VDD     1.25e+08      2.00              0          1.026e-05      4.593e-07       9.763e-05     0.0001084     1.10    INV_X32    cts_inv_529760848
  VDD      VDD     1.25e+08      2.00              0          1.146e-05      4.593e-07        9.52e-05     0.0001071     1.10    INV_X32    cts_inv_527960830
  VDD      VDD     1.25e+08      2.00              0          1.146e-05      4.593e-07       9.519e-05     0.0001071     1.10    INV_X32    cts_inv_526960820
  VDD      VDD     1.25e+08      2.00              0          9.955e-06      4.593e-07       9.606e-05     0.0001065     1.10    INV_X32    cts_inv_557661127
  VDD      VDD     1.25e+08      2.00              0          3.641e-06      2.297e-07       0.0001023     0.0001062     1.10    INV_X16    cts_inv_526660817
  VDD      VDD     1.25e+08      2.00              0          4.285e-06      2.297e-07       0.0001017     0.0001062     1.10    INV_X16    cts_inv_526160812
  VDD      VDD     1.25e+08      2.00              0          8.316e-06      4.593e-07       9.746e-05     0.0001062     1.10    INV_X32    cts_inv_530560856
  VDD      VDD     1.25e+08      2.00              0          1.012e-05      4.593e-07       9.414e-05     0.0001047     1.10    INV_X32    cts_inv_530260853
  VDD      VDD     1.25e+08      2.00              0           8.49e-06      4.593e-07       9.529e-05     0.0001042     1.10    INV_X32    cts_inv_528760838
  VDD      VDD     1.25e+08      2.00              0          4.687e-06      2.297e-07       9.916e-05     0.0001041     1.10    INV_X16    cts_inv_528460835
  VDD      VDD     1.25e+08      2.00              0          8.923e-06      4.593e-07       9.472e-05     0.0001041     1.10    INV_X32    cts_inv_588961440
  VDD      VDD     1.25e+08      2.00              0          1.129e-05      4.593e-07       9.174e-05     0.0001035     1.10    INV_X32    cts_inv_558561136
  VDD      VDD     1.25e+08      2.00              0           1.02e-05      4.593e-07        9.25e-05     0.0001032     1.10    INV_X32    cts_inv_526560816
  VDD      VDD     1.25e+08      2.00              0          8.797e-06      4.593e-07       9.374e-05      0.000103     1.10    INV_X32    cts_inv_558261133
  VDD      VDD     1.25e+08      2.00              0          8.875e-06      4.593e-07        9.35e-05     0.0001028     1.10    INV_X32    cts_inv_528060831
  VDD      VDD     1.25e+08      2.00              0          9.046e-06      4.593e-07       9.281e-05     0.0001023     1.10    INV_X32    cts_inv_590461455
  VDD      VDD     1.25e+08      2.00              0          8.415e-06      4.593e-07       9.263e-05     0.0001015     1.10    INV_X32    cts_inv_589961450
  VDD      VDD     1.25e+08      2.00              0          1.055e-05      4.593e-07        9.04e-05     0.0001014     1.10    INV_X32    cts_inv_530760858
  VDD      VDD     1.25e+08      2.00              0          1.194e-05      4.593e-07       8.842e-05     0.0001008     1.10    INV_X32    cts_inv_589461445
  VDD      VDD     1.25e+08      2.00              0          4.774e-06      2.297e-07       9.552e-05     0.0001005     1.10    INV_X16    cts_inv_589161442
  VDD      VDD     1.25e+08      2.00              0          1.057e-05      4.593e-07       8.941e-05     0.0001004     1.10    INV_X32    cts_inv_530460855
  VDD      VDD     1.25e+08      2.00              0          3.734e-06      2.297e-07       9.647e-05     0.0001004     1.10    INV_X16    cts_inv_531960870
  VDD      VDD     1.25e+08      2.00              0          1.179e-05      4.593e-07       8.766e-05       9.99e-05     1.10    INV_X32    cts_inv_529460845
```

Scaling Power

- **Why scale power ?**
  - What if the power from user defined toggle rates in scenario view doesn't matches design target power ?
  - Can scale power from an input power view to the target power / target toggle_rate
  - Can create multiple scaled power view
- **How to scale power ?**
  - Target Power based
  - Target Toggle Rate based
  - Leakage Power scaling
  - Exclude scaling for specific power source like IPF , ModeControl etc via disable_scaling_sources
  - Disable scaling on a specific scope
  - For more details , see : **help(SeaScapeDB.create_scaled_power_view)**



Scaling Power

File : scripts/args.py

```
pwr_mixed_scaled_args = dict(
    target_power_settings=target_power_settings,
    tag='pwr_mixed_scaled',
    object_settings = scaled_power_object_settings,
    options=options)
target_power_settings = { Instance('') : {Net('VDD') : { 'target_power' : 0.15 },
                                          Net('core3/VDD_INT') : { 'target_power' : 0.0332 }
                                         },}
scaled_power_object_settings = {'design_values' : { 'disable_scaling_sources' : ['modecontrol'] ,},}
```

File : scripts/static_analysis.py

```
pwr_mixed_scaled = db.create_scaled_power_view(pwr_orig=pwr_mixed,**pwr_mixed_scaled_args ) reports_dir =
'reports/powercalc'

gp_util.makedir(reports_dir)

emir_reports.write_instance_power_report_and_summary(pwr_mixed_scaled, reports_dir+'/instance_power_mixed_scaled.rpt',
reports_dir+'/power_summary_mixed_scaled.rpt')
```

Looking At Scaled Power View Reports

File : power_summary_mixed_scaled.rpt

**** RedHawk-SC Power Summary Report ****
Created: Sun Sep 20 21:19:37 2020

A total of 1185384 instances were summarized for this report while 0 were omitt

A total of 0 pins were omitted because they were not attached to a power domai

***

VDD Domain total power is now 0.15 and core3/VDD_INT is 0.0332 as set in target_power_settings of scaled power view

grouping	clock_pin_power(W)	internal_power(W)	leakage_power(W)	switching_power(W)	total_power(W)	percent_power(%)	pin_count	instance_count
*** Power Domains:								
VDD	0.029815	0.054228	0.011256	0.084516	0.15	81.88	912472	909926
core3/VDD_INT	0.0028889	0.010238	0.0035269	0.019435	0.0332	18.12	275458	275458
Total	0.032704	0.064466	0.014782	0.10395	0.1832	100.00	1187930	1185384
*** Frequency Domains:								
1.25e+08	0.032704	0.064466	0.013739	0.10395	0.18216	99.43	377445	377445
0	0	2.9262e-08	0.0010435	0	0.0010435	0.57	810485	807939
Total	0.032704	0.064466	0.014782	0.10395	0.1832	100.00	1187930	1185384
*** User Defined Groups:								
combinational logic	0	0.017603	0.0090181	0.095338	0.12196	66.57	310766	308220
sequential logic	0.032204	0.04637	0.0057205	0.0086055	0.060696	33.13	71800	71800
memory	0.00049972	0.00049296	4.3939e-05	6.9543e-06	0.00054366	0.30	8	8
decap/filler	0	0	0	0	0	0.00	805356	805356
Total	0.032704	0.064466	0.014782	0.10395	0.1832	100.00	1187930	1185384

**** End Report ****

Memory total power remained the same as their power scaling was disabled in scaled power view via object_settings

Static Voltage Drop Analysis

Performing Static Analysis

Static ScenarioView

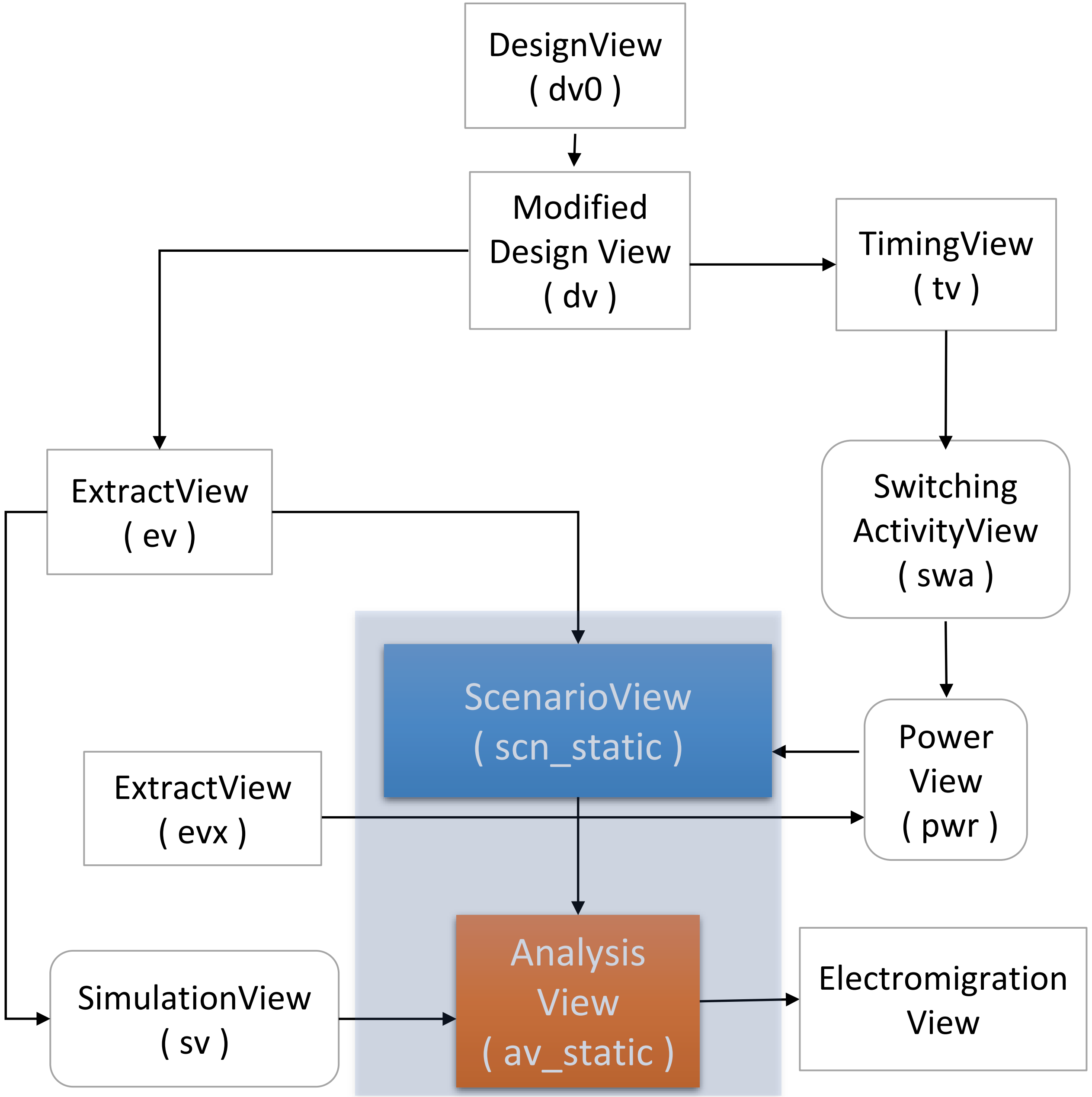
- Converts the power per instance to DC current per PG pin
- For more details , see :

  help(SeaScapeDB.create_scenario_view)

AnalysisView

- Holds results of simulation
- Takes in SimulationView and ScenarioView as inputs
- All voltage and current queries are available after simulation from AnalysisView
- For more details , see :

  help(SeaScapeDB.create_analysis_view)



Preparing for static run

File : scripts/args.py

```
scn_mixed_static_args = dict(

    voltage_levels=voltage_levels,

    scenario_type='Static',

    tag='scn_mixed_static',

    options=options)

av_mixed_static_args = dict(

    tag='av_mixed_static',

    keep_stats_level='Full',

    options=options)
```

File : scripts/static_analysis.py

```
scn_mixed_static = db.create_scenario_view(power_view=pwr_mixed_scaled,
**scn_mixed_static_args)

av_mixed_static = db.create_analysis_view(simulation_view=sv,
scenario_views=scn_mixed_static, **av_mixed_static_args)
```

Static ScenarioView

- Reads in voltage levels for each Domain and Converts the power per instance to DC current per PG pin
- Pin vdd total power of `0.0001734` W is converted to current of `0.0001734/1.1 = 1.577e-4` in Static Scenario View

```
pwr_mixed.get_attributes(Instance("core3/program_memory")
)

{Pin('vdd'): {'clock_pin_power': 0.00016499549383297563,

             'frequency': 125000000.0,

             'internal_power': 0.00016346415213774 89,

             'leakage_power': 8.498927854816 43e-06,

             'source': 'ModeControl',

             'switching_power': 1.5313393078031368e-06,

             'toggle_rate': 0.6099999547004 7,

             'total_power': 0.00017349442350678146,

             'voltage': 1.100000023841858}}
```

```
scn_mixed_static.get_demand_current(Instance("core3/program_memo
ry"),Pin('vdd'))

Waveform([

(0.0, 0.0001577221992 19279),

])

scn_mixed_static.get_demand_current(Instance("core3/program_memo
ry"),Pin('gnd'))

Waveform([

(0.0, -0.000157722199219279),
```

Looking At Static Analysis Reports

File : scripts/static_analysis.py

```
reports_dir = 'reports/static_analysis'
gp_util.makedir(reports_dir)

emir_reports.write_all_instance_voltages(av_mixed_static, reports_dir+'/inst_voltage.rpt')
emir_reports.write_bump_currents(av_mixed_static, reports_dir+'/bump_current.rpt')
emir_reports.write_bump_voltages(av_mixed_static, reports_dir+'/bump_voltage.rpt')
emir_reports.write_demand_currents(av_mixed_static, reports_dir+'/demand_current.rpt')
emir_reports.write_layer_voltage_report(av_mixed_static, reports_dir+'/layer_voltage.rpt')
emir_reports.write_node_voltage_report(av_mixed_static, reports_dir+'/node_voltage.rpt')
emir_reports.write_supply_currents(av_mixed_static, reports_dir+'/supply_currents.rpt')
emir_reports.write_switch_report(av_mixed_static, reports_dir+'/switch_report.rpt')
```

Static Analysis Reports are available In 'reports/static_analysis' directory

```
inst_voltage.rpt
demand_current.rpt
bump_current.rpt
bump_voltage.rpt
layer_voltage.rpt
node_voltage.rpt
supply_currents.rpt
switch_report.rpt
```

Looking At Static Analysis Reports

```
TitleText: Bump Currents
# Time (ps) I (A)
"VSS_1
        0.00    -0.0000010

"VSS_10
        0.00    -0.0004140

"VSS_100
        0.00    -0.0008939

"VSS_101
        0.00    -0.0002740

"VSS_102
        0.00    -0.0006438
```

```
TitleText: Bump Voltages
# Time (ps) V (V)
"VSS_1
        0.00     0.0000000

"VSS_10
        0.00     0.0000000

"VSS_100
        0.00     0.0000001

"VSS_101
        0.00     0.0000000

"VSS_102
        0.00     0.0000001
```

bump_current.rpt / bump_voltage.rpt

```
# contents are sorted in decending order by layer, net, node_voltage
# loc_x     loc_y     layer      net     node_voltage
#  (u)       (u)                           (v)
  1088.4    64.8     metal12     VSS     5.391e-05
  1083.6    64.8     metal12     VSS     5.327e-05
  1088.4    63.2     metal12     VSS     5.193e-05
  1086.8    64.8     metal12     VSS     5.169e-05
  1085.2    64.8     metal12     VSS     5.145e-05
  1083.6    63.2     metal12     VSS     5.128e-05
  1088.4    66.4     metal12     VSS      4.98e-05
  1083.6    66.4     metal12     VSS     4.977e-05
  1086.8    66.4     metal12     VSS     4.971e-05
```

node_voltage.rpt

```
# loc_x      loc_y      min_instance_static_voltage   pg_arc              instance
#  (u)        (u)                  (v)                pwr/gnd
 1196.265   221.305               1.098               core3/VDD_INT/VSS core3/openMSP430_inst1.clock_module_0.clock_gate_dbg_clk.CLKGATETST_X1_1
 1214.505   221.305               1.098               core3/VDD_INT/VSS core3/_14250_
 1210.515   221.305               1.098               core3/VDD_INT/VSS core3/_24170_
 1206.715   221.305               1.098               core3/VDD_INT/VSS core3/_24128_
 1206.525   221.305               1.098               core3/VDD_INT/VSS core3/HFSBUF_352_3764
 1201.775   221.305               1.098               core3/VDD_INT/VSS core3/_24139_
 1196.265   221.305               1.098               core3/VDD_INT/VSS core3/_23837_
 1194.555   221.305               1.098               core3/VDD_INT/VSS core3/_23840_
 1194.555   221.305               1.098               core3/VDD_INT/VSS core3/_23841_
 1193.035   221.305               1.098               core3/VDD_INT/VSS core3/_23834_
 1192.845   221.305               1.098               core3/VDD_INT/VSS core3/HFSBUF_5391_328
```

inst_voltage.rpt

```
#  min_x     min_y    min_voltage_drop   max_x      max_y    max_voltage_drop  layer_drop  net            layer
#   (u)       (u)          (v)            (u)        (u)          (v)            (v)
 1001.56    156.905    6.891e-05      1225.98     603.4       0.001351      0.001282   VSS            metal1
 960.885    189.165    0.0001897      1197.8675   222.645     0.0009766     0.000787   core3/VDD_INT  metal1
 1001.685   156.905    6.818e-05      1206.525    159.075     0.0006576     0.0005894  VSS            metal2
 1002.485   156.905    6.718e-05      1182.485    224.105     0.0003217     0.0002545  VSS            metal3
 999.285    156.8      5.313e-05      1180.885    221.8        0.000307     0.0002538  VSS            metal10
 1002.235   156.905    6.594e-05      1182.485    224.105     0.0003197     0.0002538  VSS            metal4
 1002.235   156.905    6.544e-05      1182.685    224.105     0.0003186     0.0002532  VSS            metal5
```

layer_drop.rpt

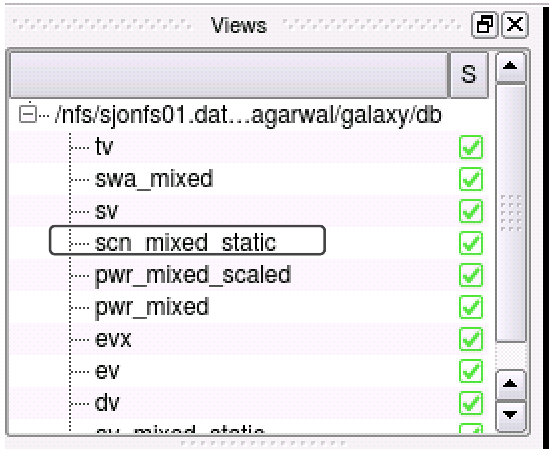
```
TitleText: Supply Currents
# Time (ps) I (A)

"VSS
        0.00    -0.1384887

"VDD
        0.00     0.1394365
```

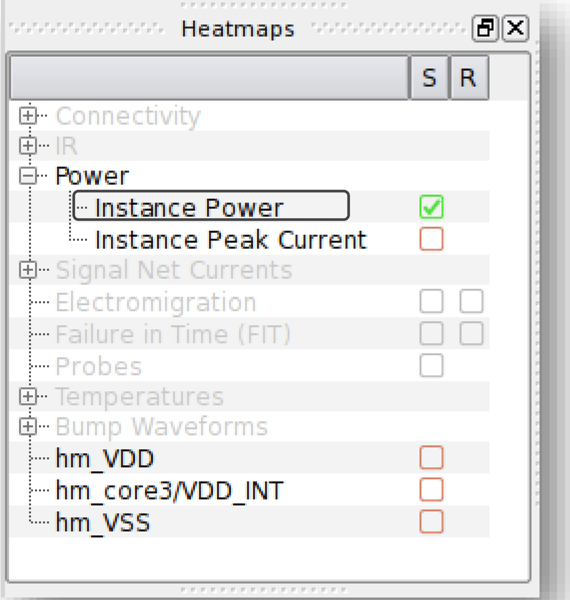
```
TitleText: Demand Currents
# Time (ps) I (A)

"core3/VDD_INT
        0.00     0.0247338

"VSS
        0.00    -0.1384887

"VDD
        0.00     0.1394365
```

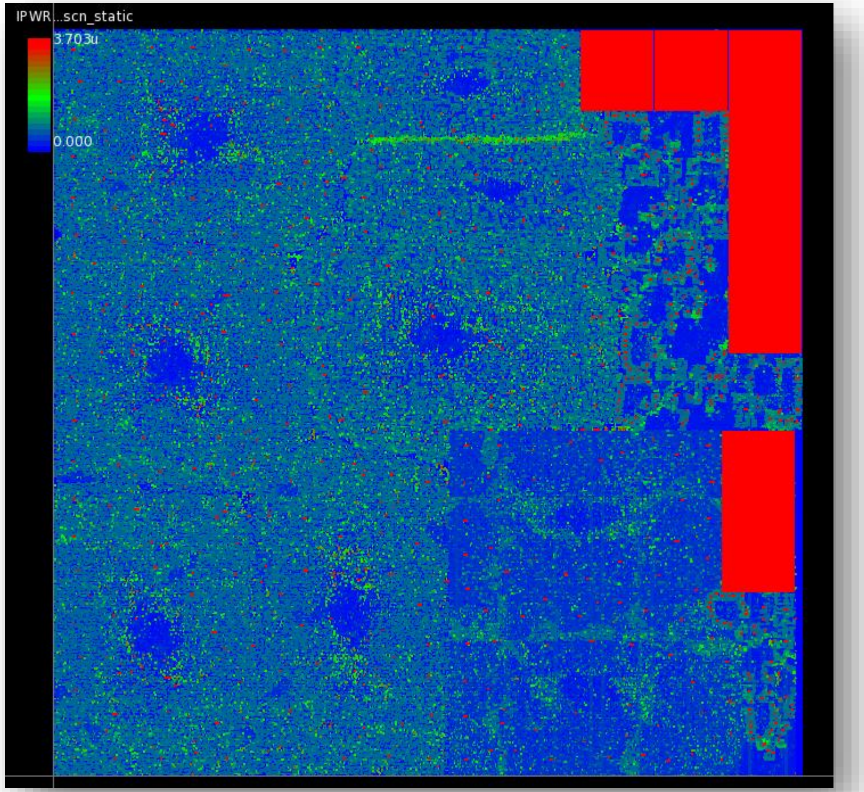
Battery & Demand Currents

Ansys

Examining Static Results In GUI

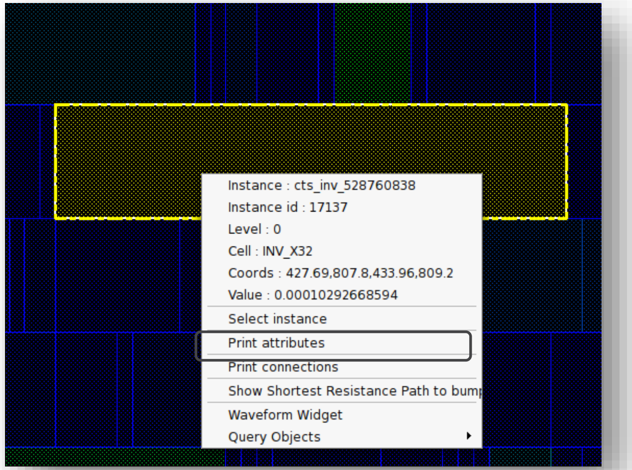


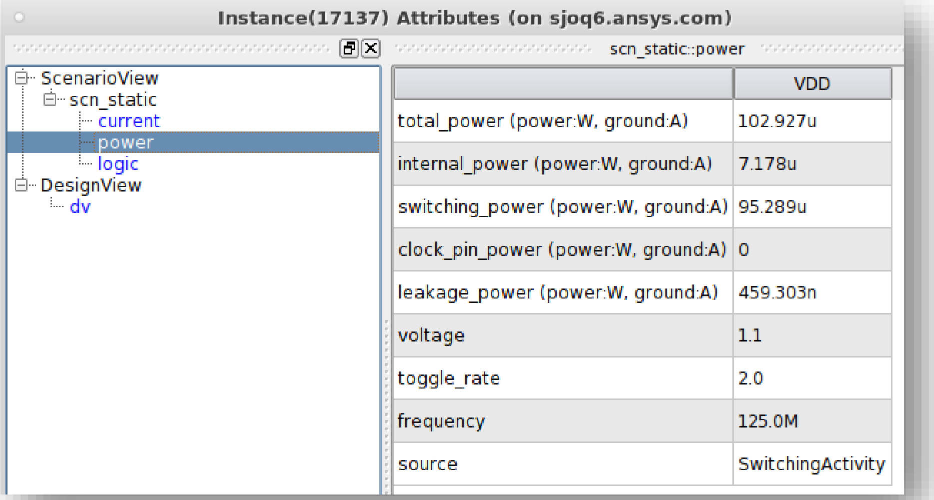




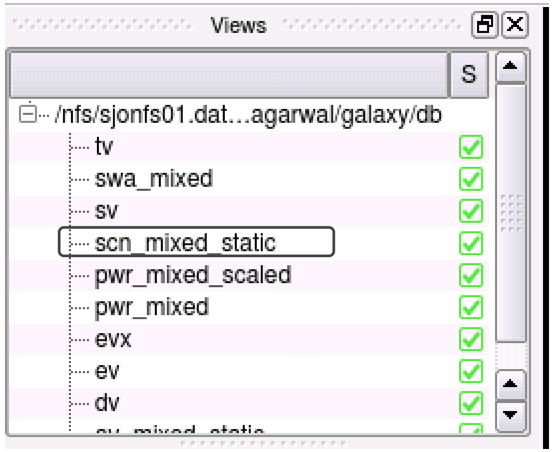
Instance Power Map

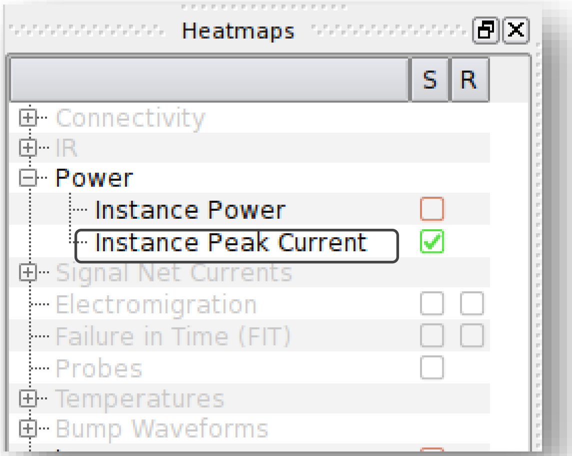
Power Attributes Query

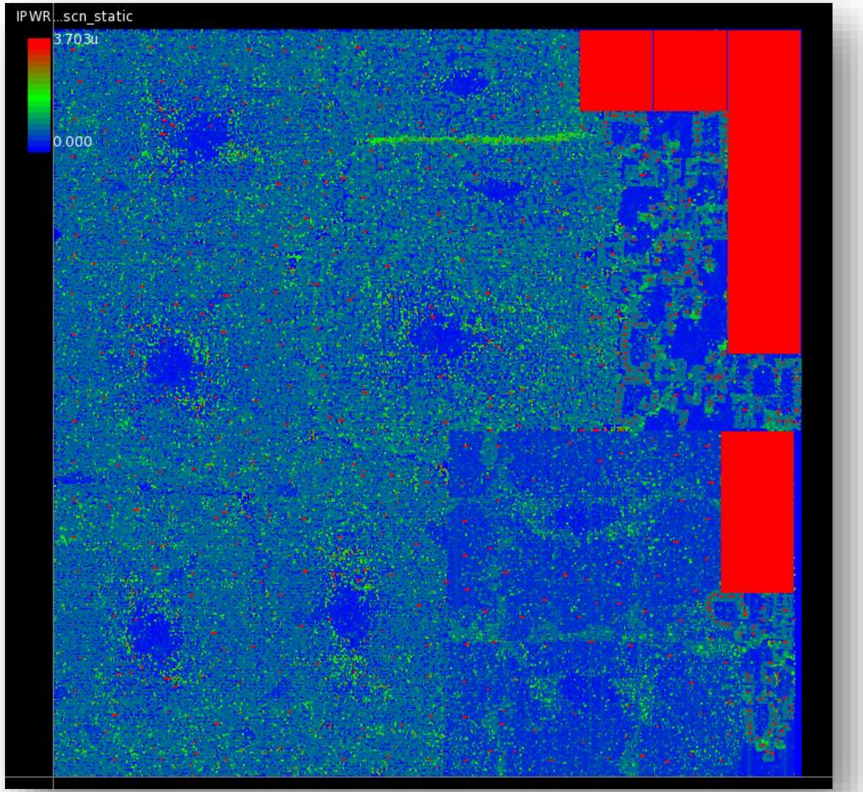




Examining Static Results In GUI

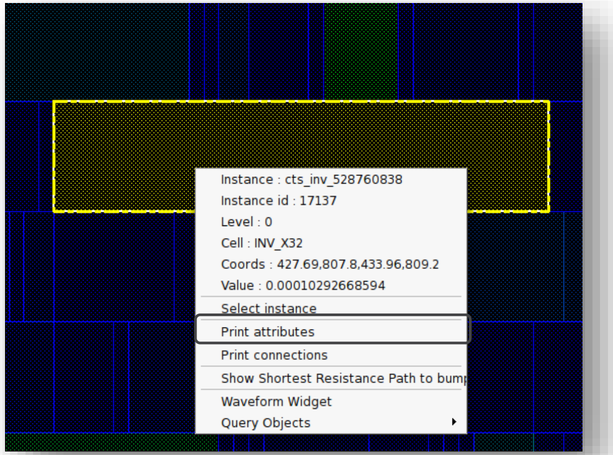


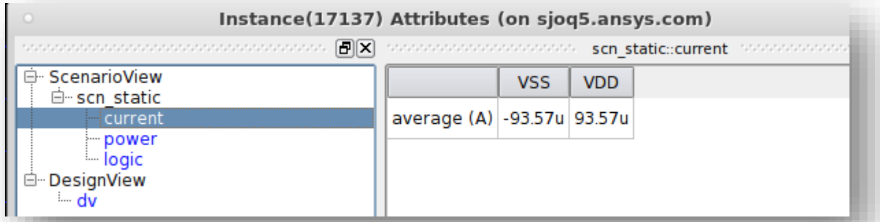




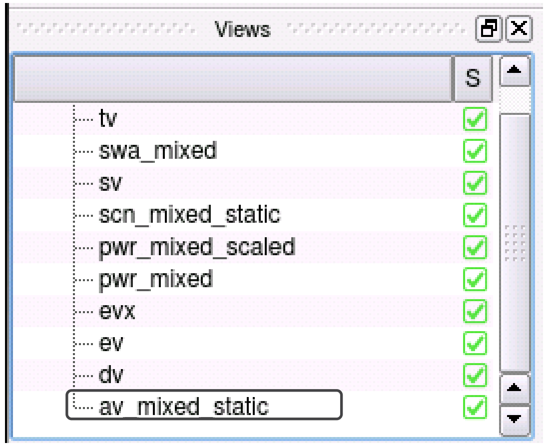
Instance Avg Current Map

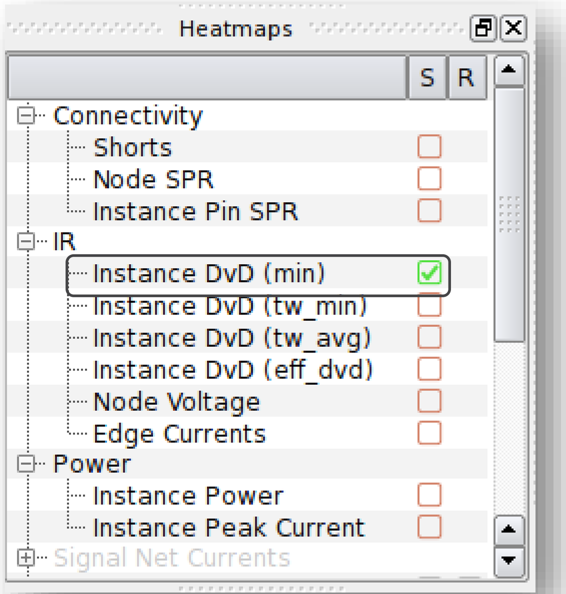
Avg Current Query

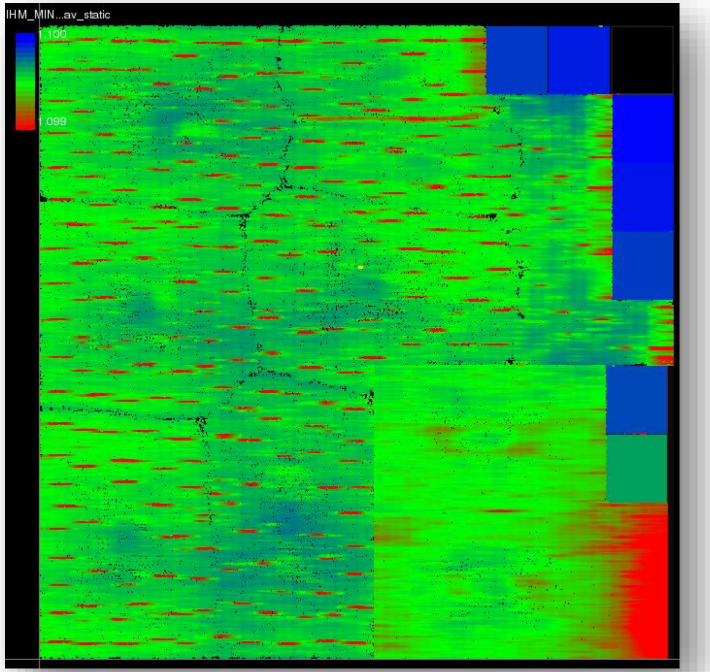




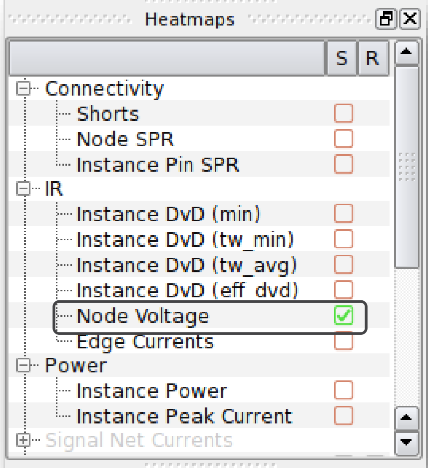
Examining Static Results In GUI

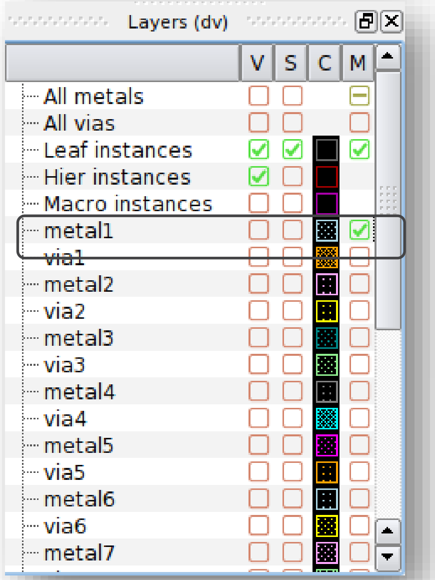




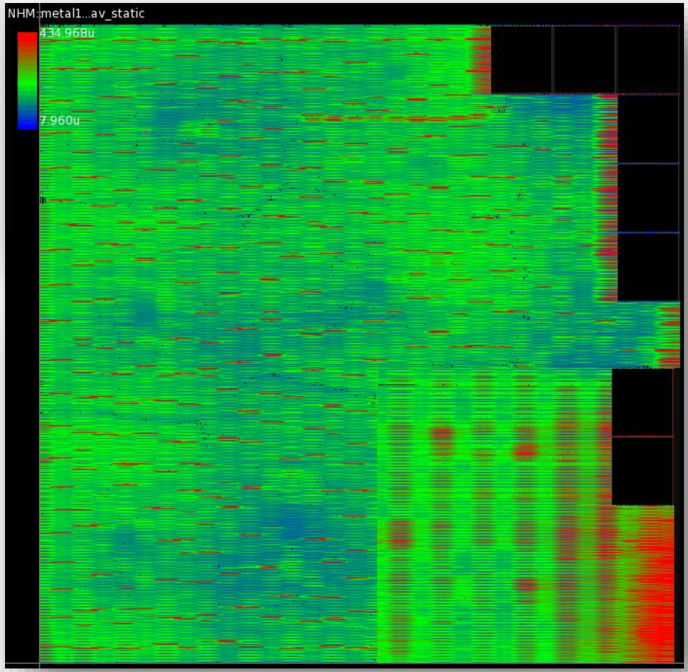


Instance Voltage Drop Map

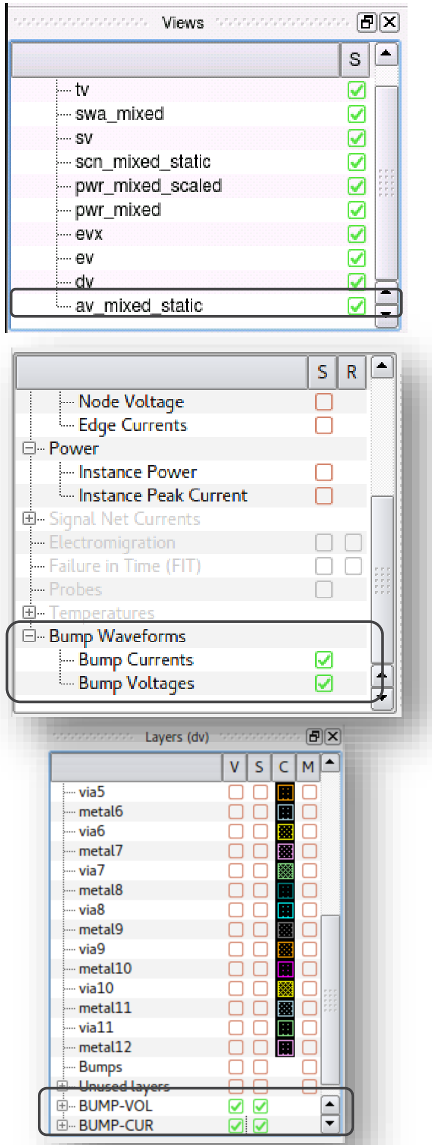


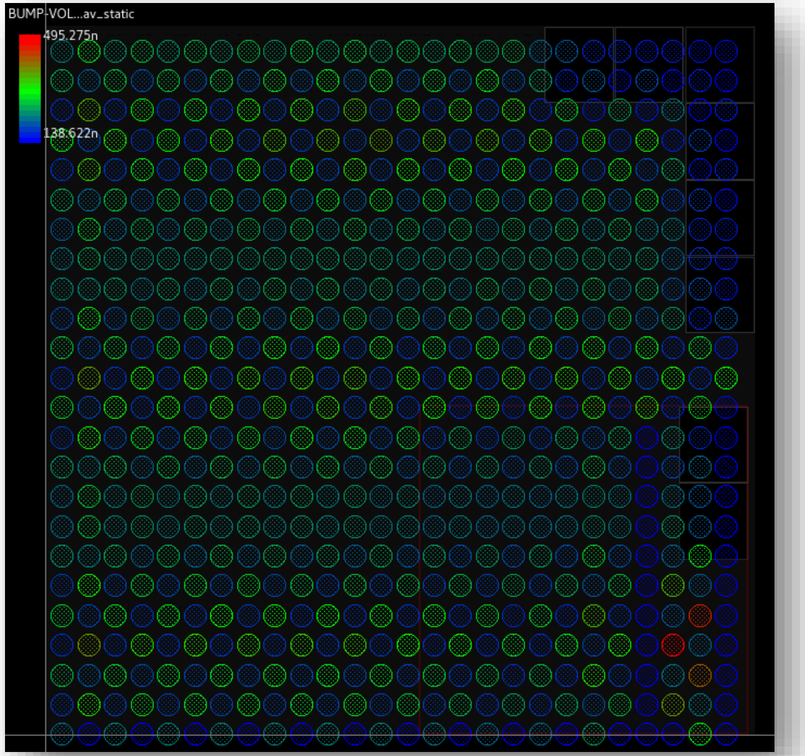


Node Voltage Drop Map
(at metal1)

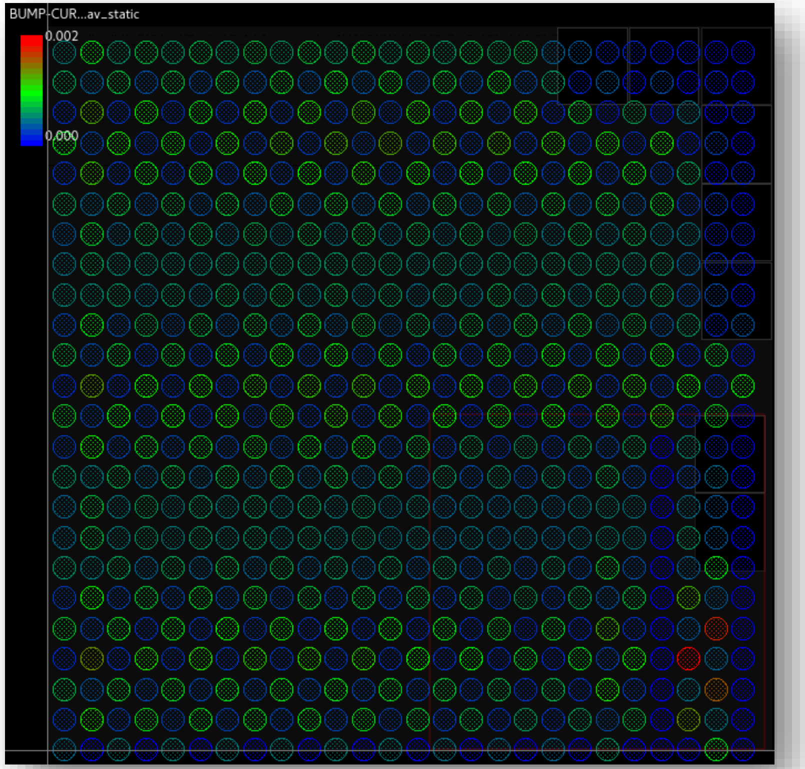


Examining Static Results In GUI



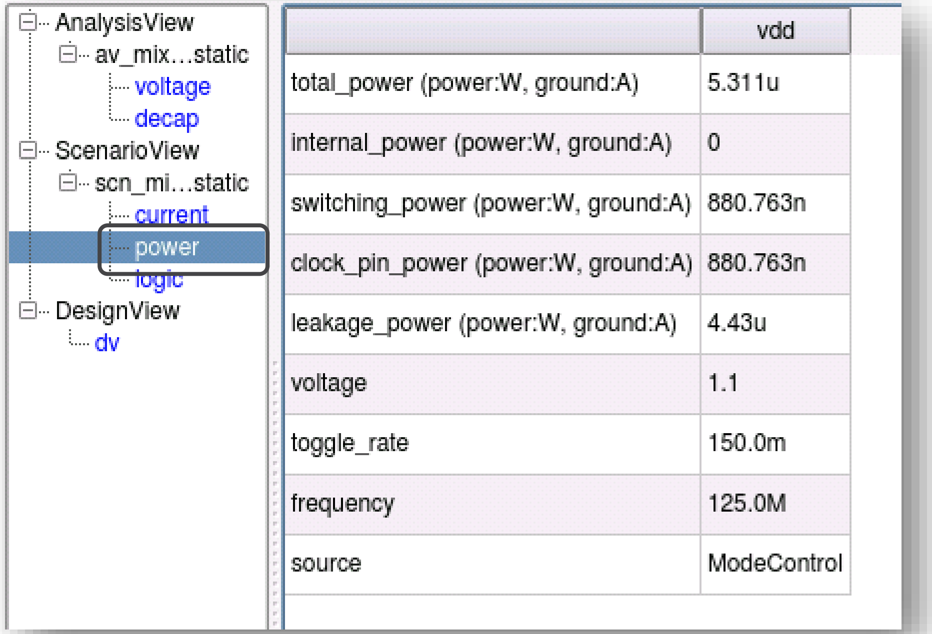


Bump Voltage Map

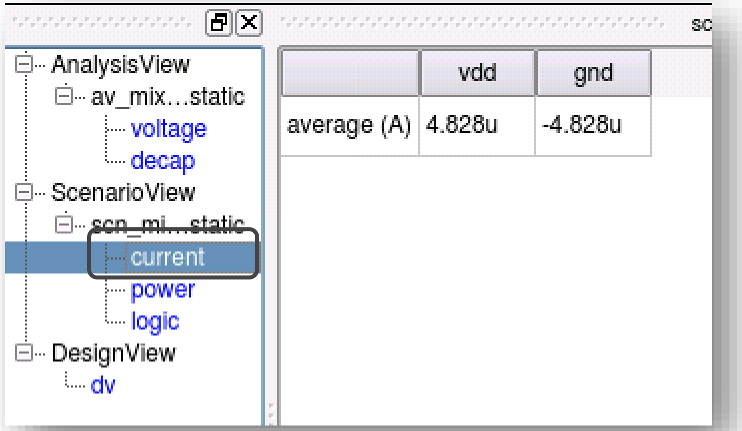


Bump Current Map

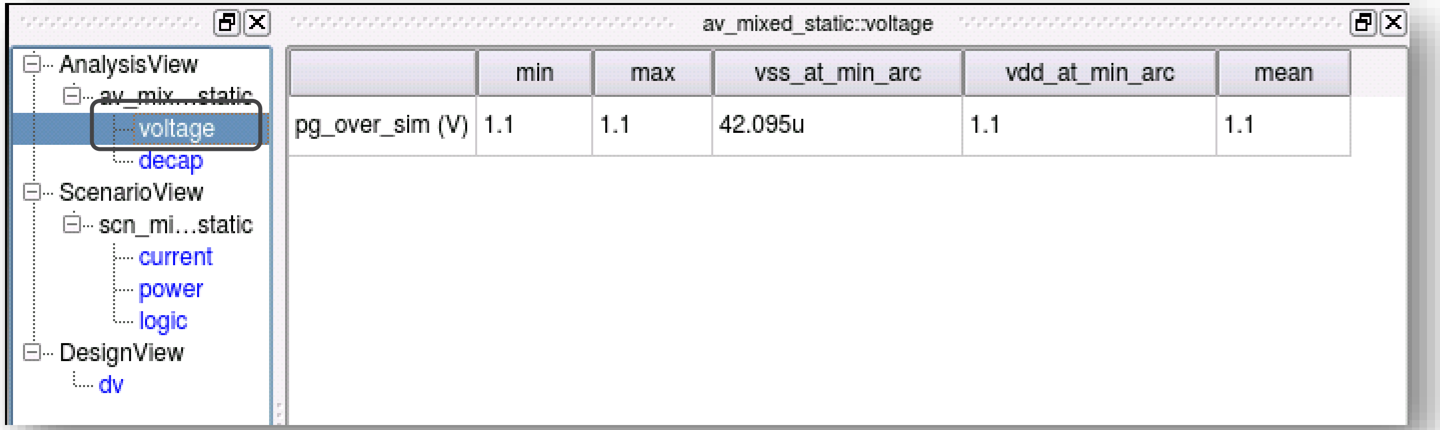
Querying Instance level power/current/voltage from GUI



Instance Power

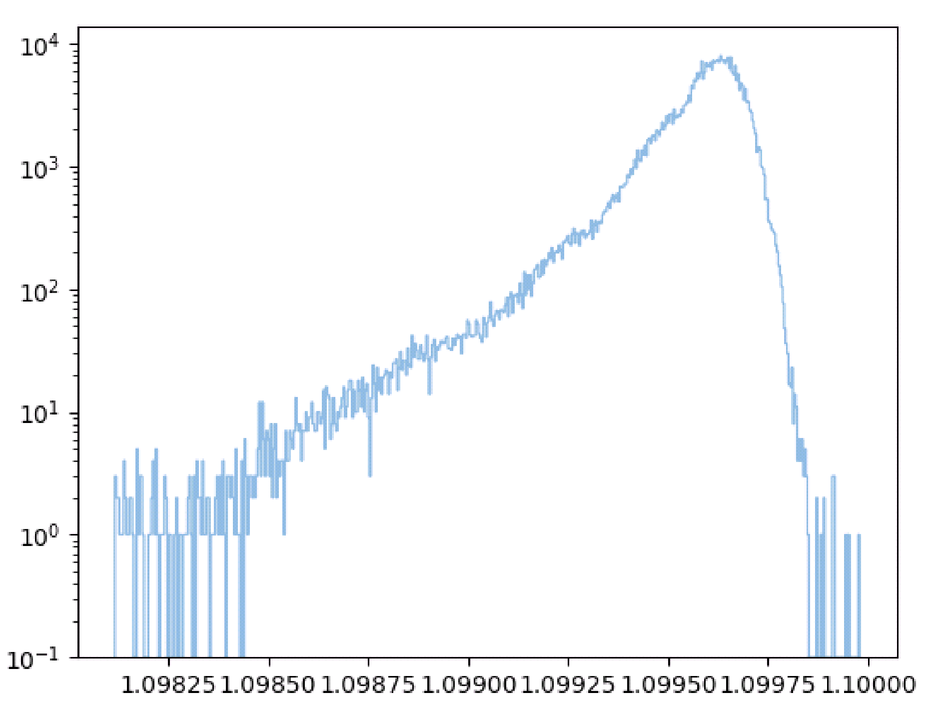


Instance Current

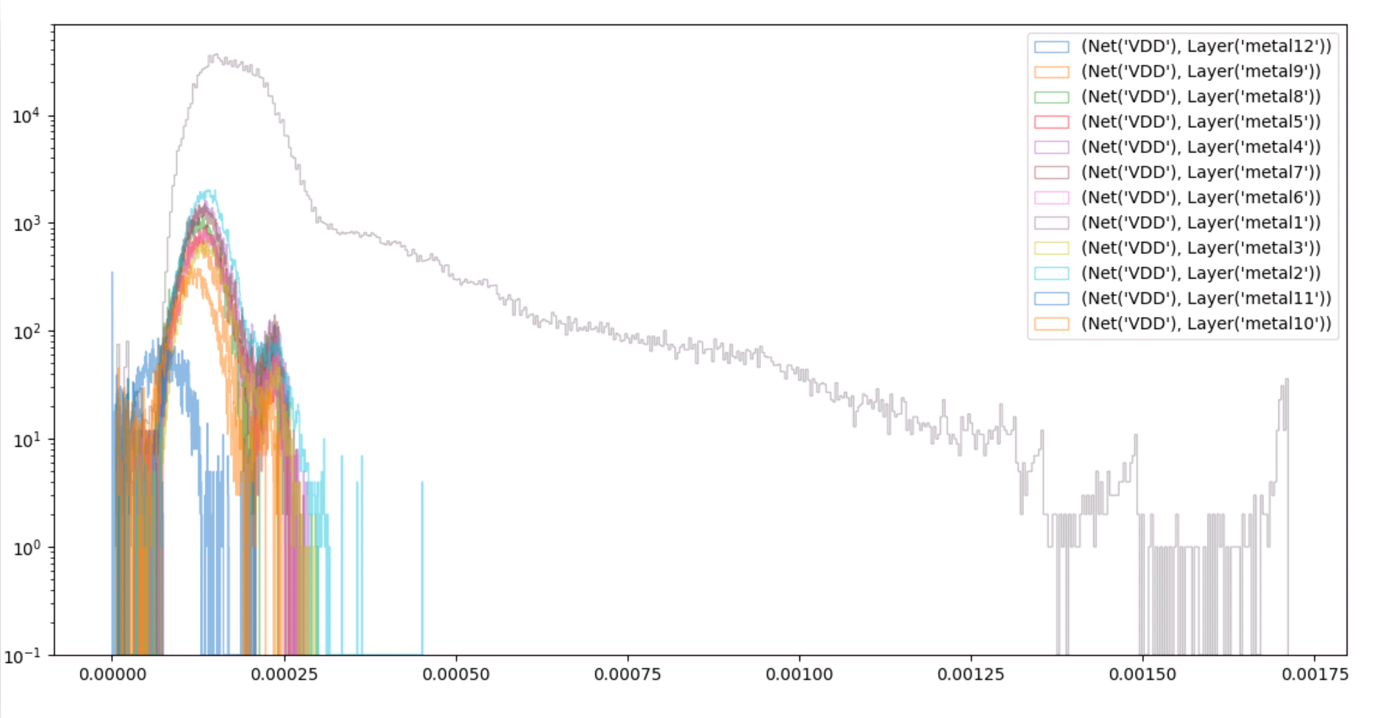


Instance Static IR

Static IR – Instance and Node Voltage Histogram

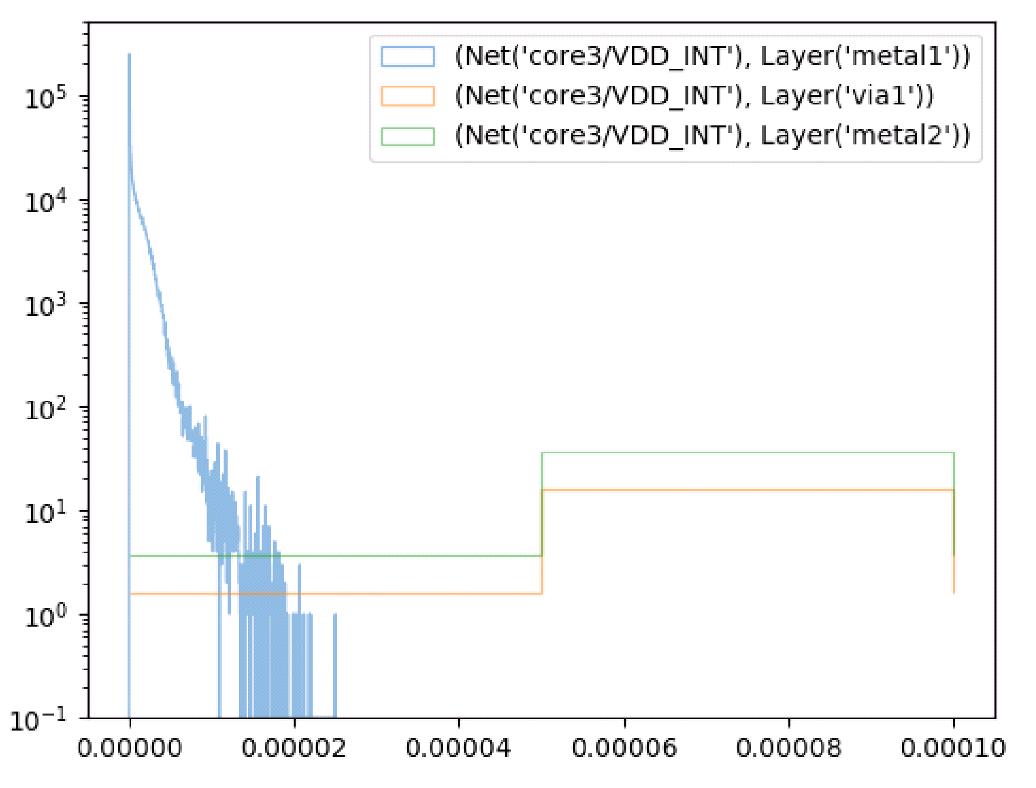


Instance Voltage Histogram

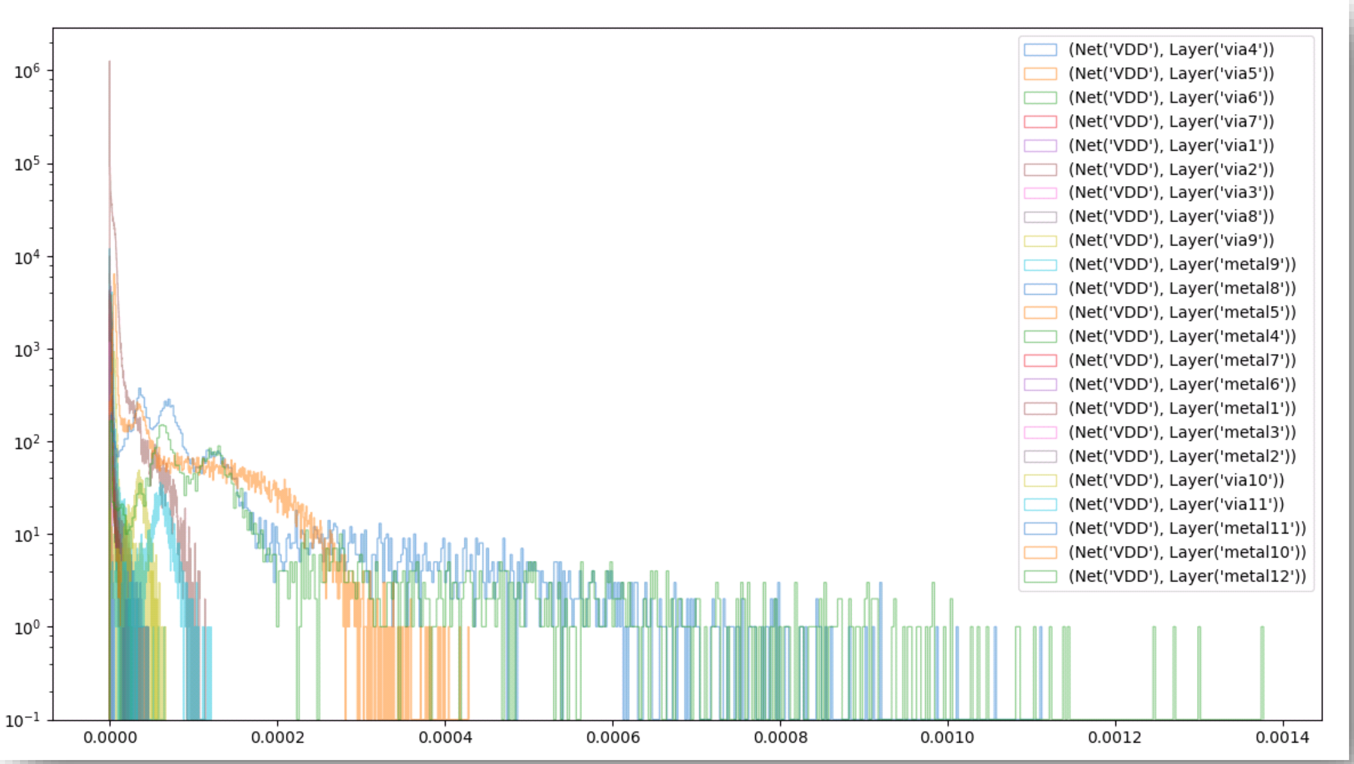


Node Voltage Histogram for VDD Domain

Current Histogram for different Domains and Layers

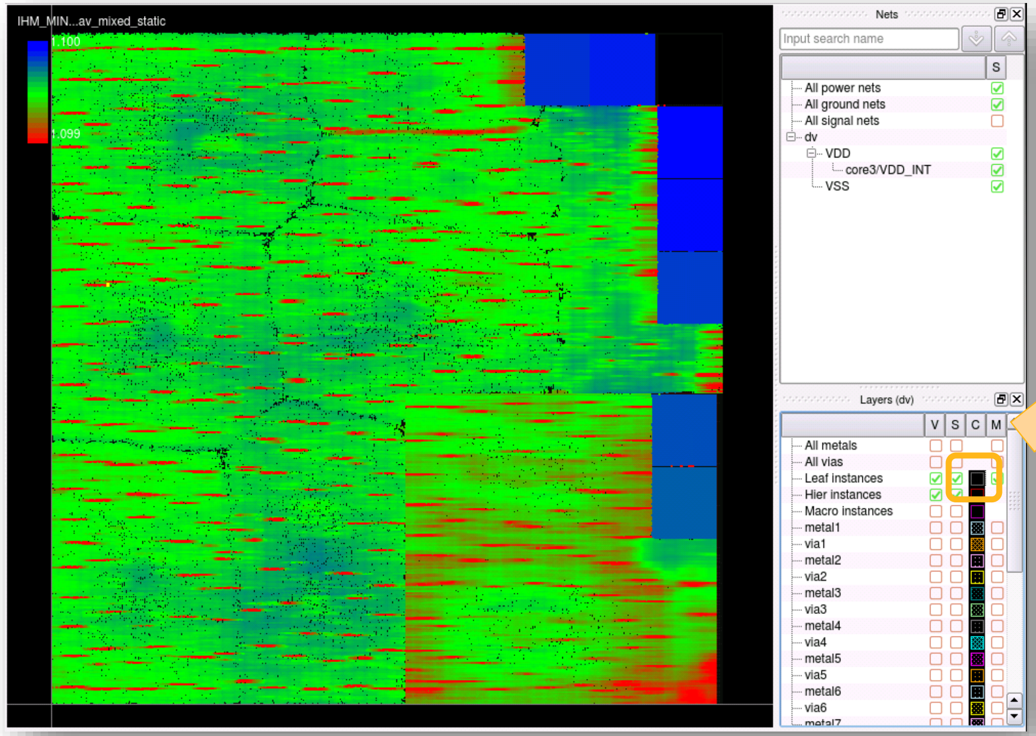


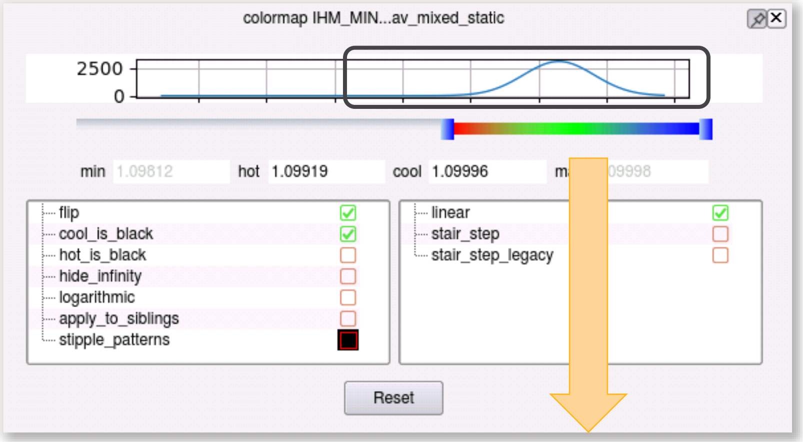
Current Histogram for Net core3/VDD_INT



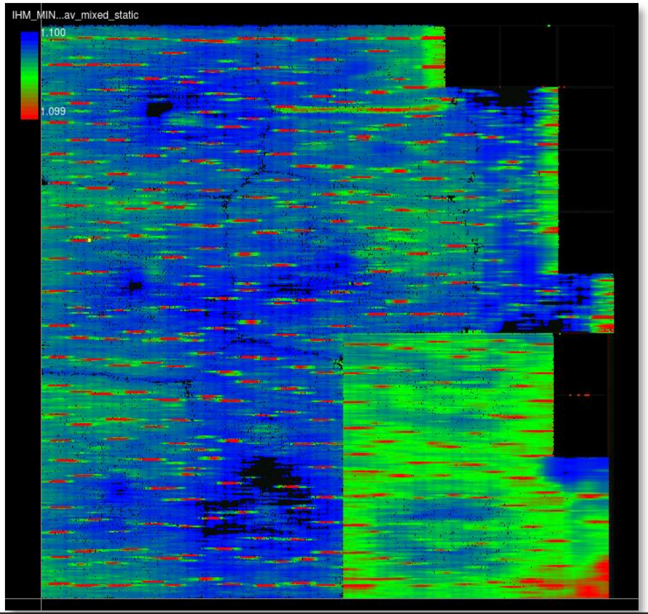
Current Histogram for Net VDD

Adjusting Color Legend in HeatMaps

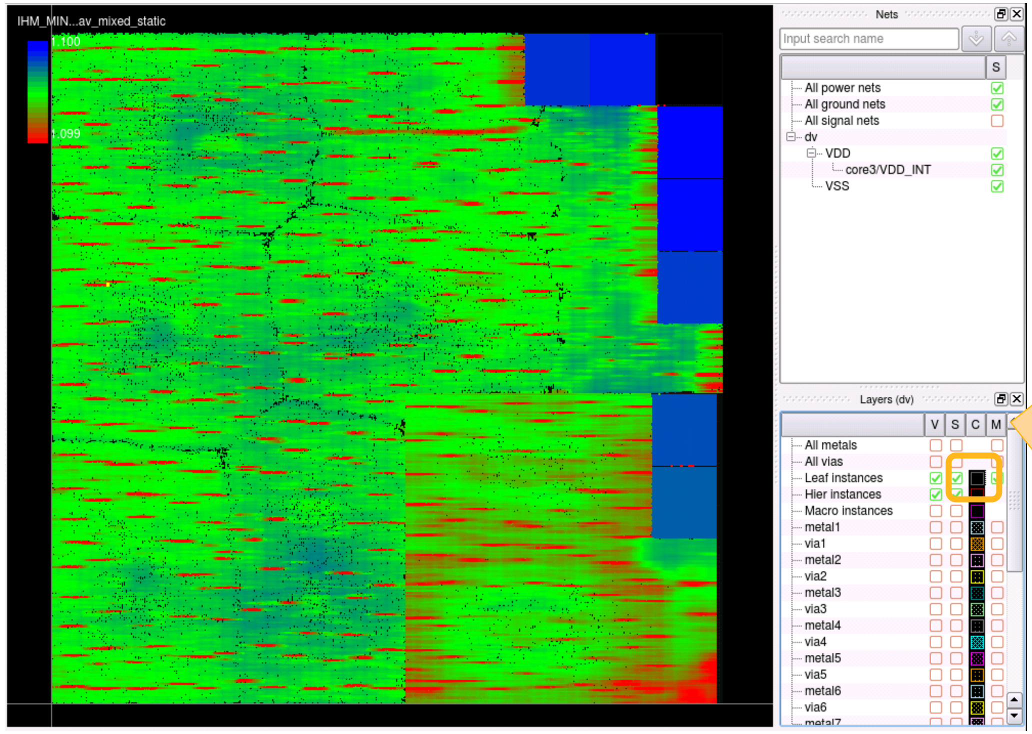




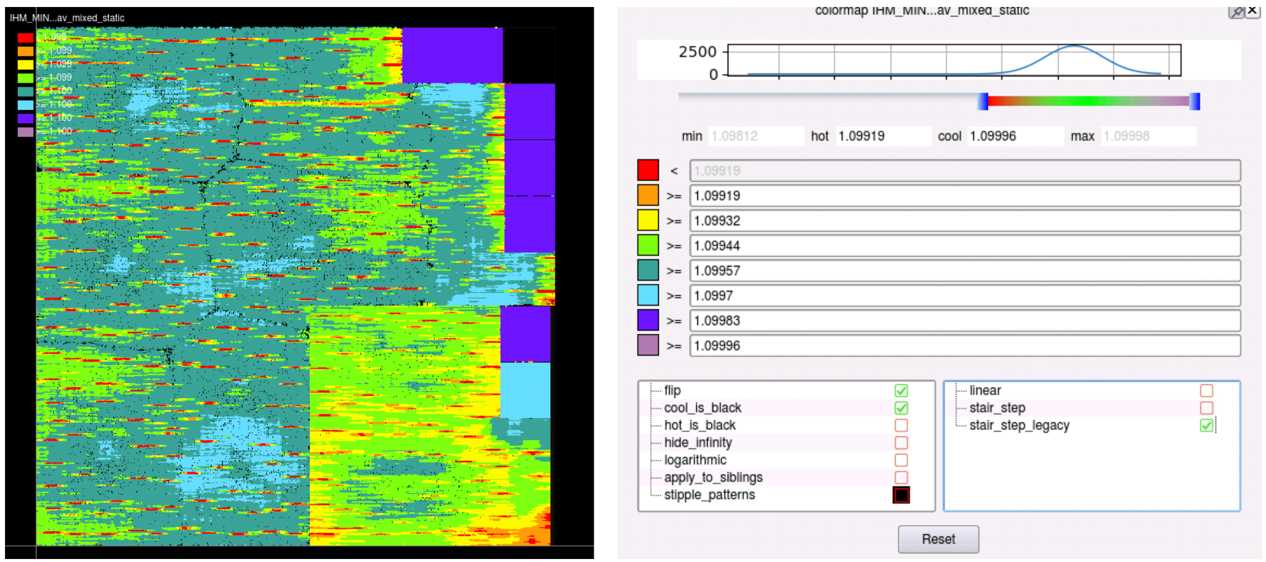
Range slider can be moved using mouse to set max and min range



Legacy Style HeatMaps



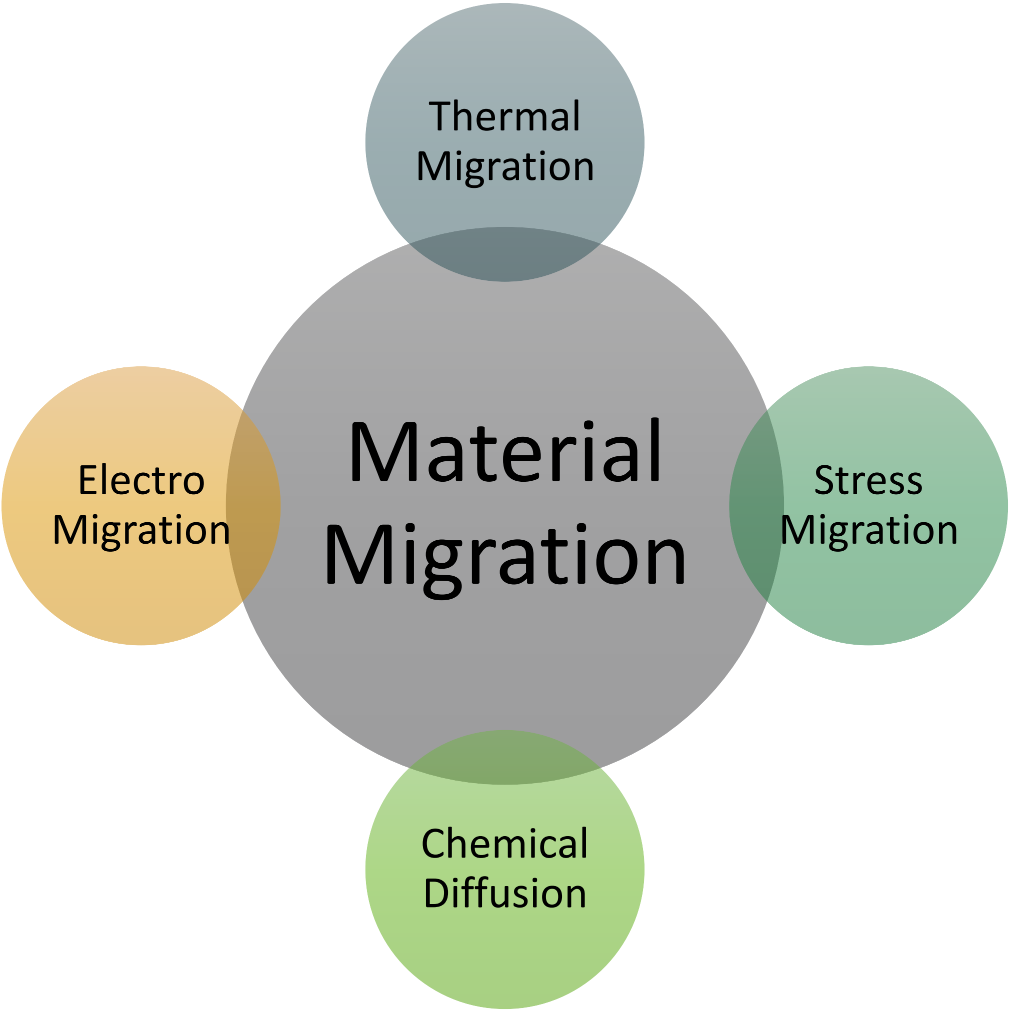
Stair Step Legacy can be selected to show legacy style HeatMaps



Basics of Electromigration

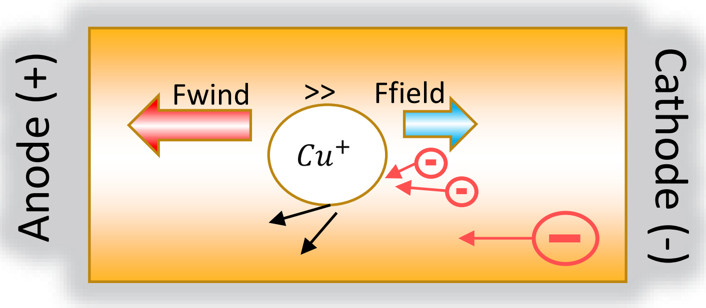
Material Migration

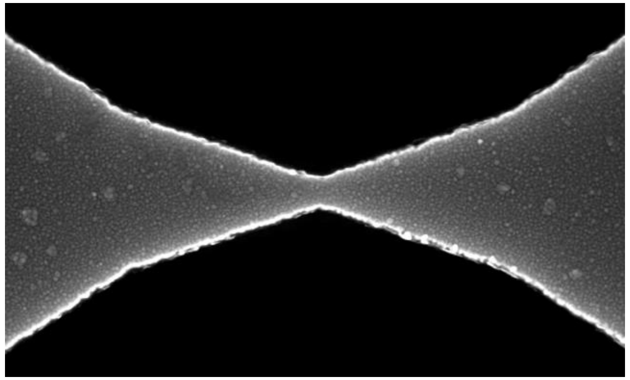
- Thermal Migration due to temperature gradients
- Stress Migration due to Mechanical Stress
- Chemical diffusion due to concentration gradients
- Electromigration due to Electrical field



Basics of Electromigration

- Current flow produces two forces
- Electrostatic Force F_{field}
  - Caused by electric field strength in the metallic conductor
- Fwind
  - Generated by the momentum transfer between the electrons and metal ions
- If this force exceeds the trigger/activation energy, material migration starts in the direction of the electron flow
- Depending on the current density, metal ion may drift instantly or eventually in time
- Wires shorter than the “Blech Length” will allow higher currents, since the mechanical effect counteracts the EM effects

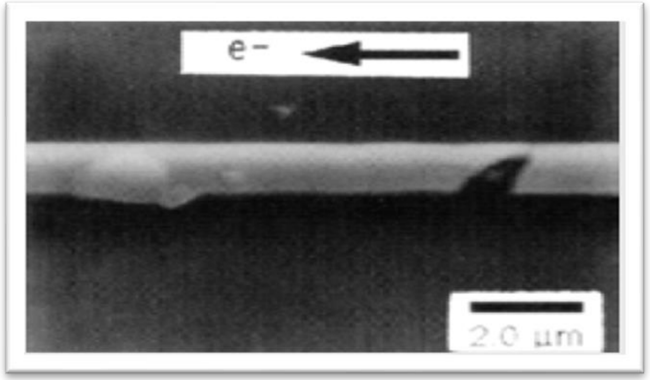


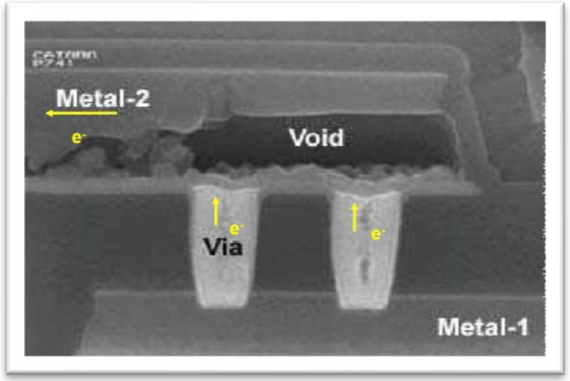


Source: Wikipedia Electromigration

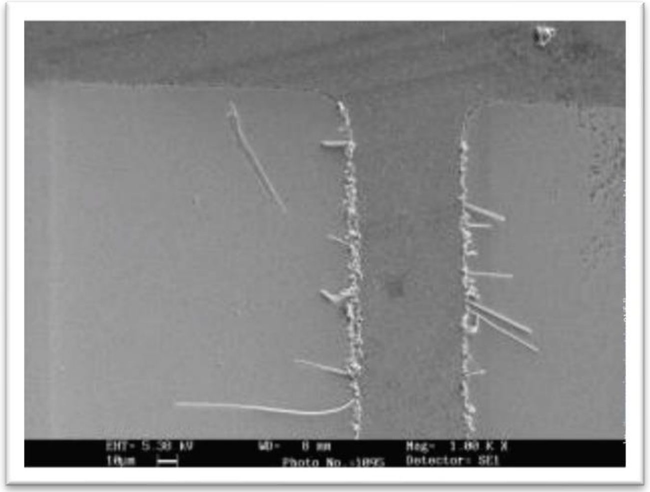
Effects of Electromigration

- Voids
  - Also known as opens
  - Metal ions are pushed out in the direction of electron flow
  - Outgoing ion flux > Incoming Ion flux
  - Open circuits leading to logic failures





- Hillocks
  - Also known as whiskers/shorts
  - Metal ions are deposited
  - Outgoing ion flux < Incoming ion flux
  - Can lead to shorts with neighboring interconnects



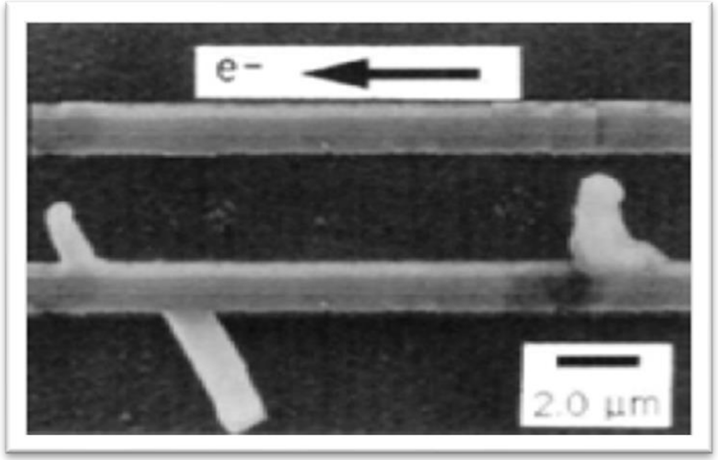
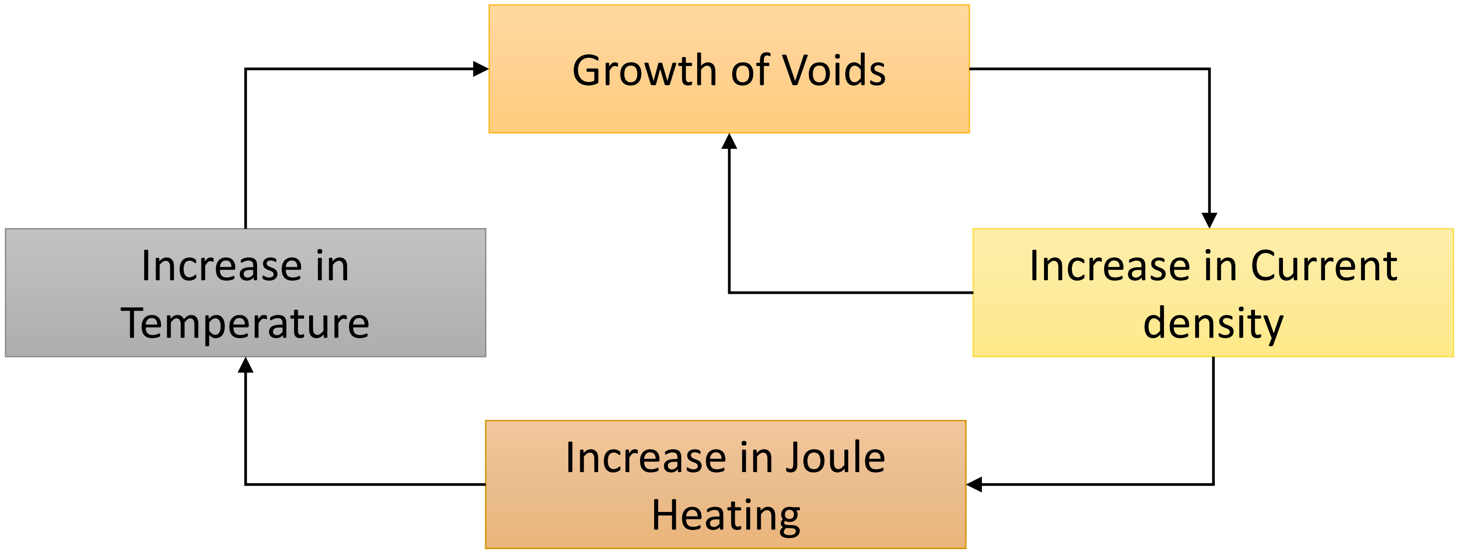


Image Source: W.D Nix et al. 1992

Effects of Electromigration

- Accelerated growth of voids by positive feedback
- Void growth increases current density
- Increasing wire temperature due to Joule heating
- Accelerates the diffusion and voids



Black's Equation, Quantifying EM

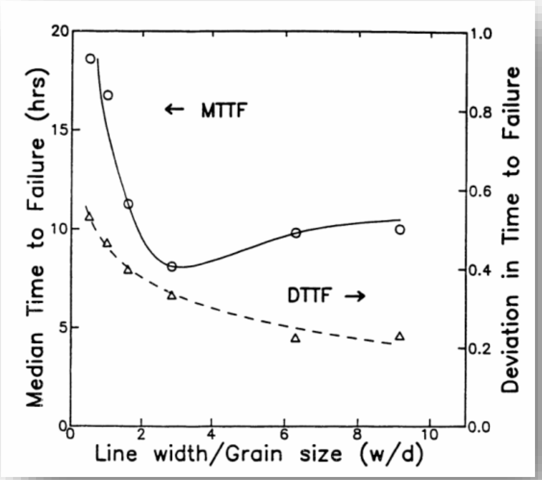
- Formula from James Black to calculate time to failure
- Explored & pinned around 1969
- Silicon devices have continued to get smaller since then
- Additional Modelling is required for this failure mechanism

$$\text{MTTF} = AJ^{-n}e^{E_a/kT}$$

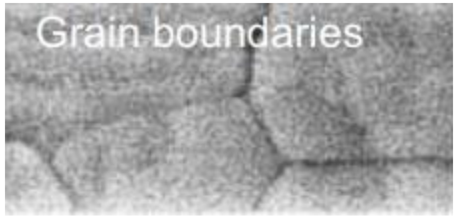
MTTF	Median Time to Failure
A	Constant related to conductor geometry
J	Current Density
n	Parameter related to current density accounting for current flow effects other than Joule heating
Ea	Activation Energy
k	Boltzmann's constant (8.61 x 10^-5 eV/k)
T	Temperature in Kelvin, K

Constraints Affecting EM

- Current Density
- Temperature
- Interconnect Geometry
  - Width
  - Length
    - Blech length (Length < blech length, mechanical stress counteracts the EM effect)
    - Here, a mechanical stress buildup causes an atom back flow process which reduces or even compensates the effective material flow towards the anode.
- Interconnect Material



Source: EM failure and Reliability. YOUNG-CHANG JOO



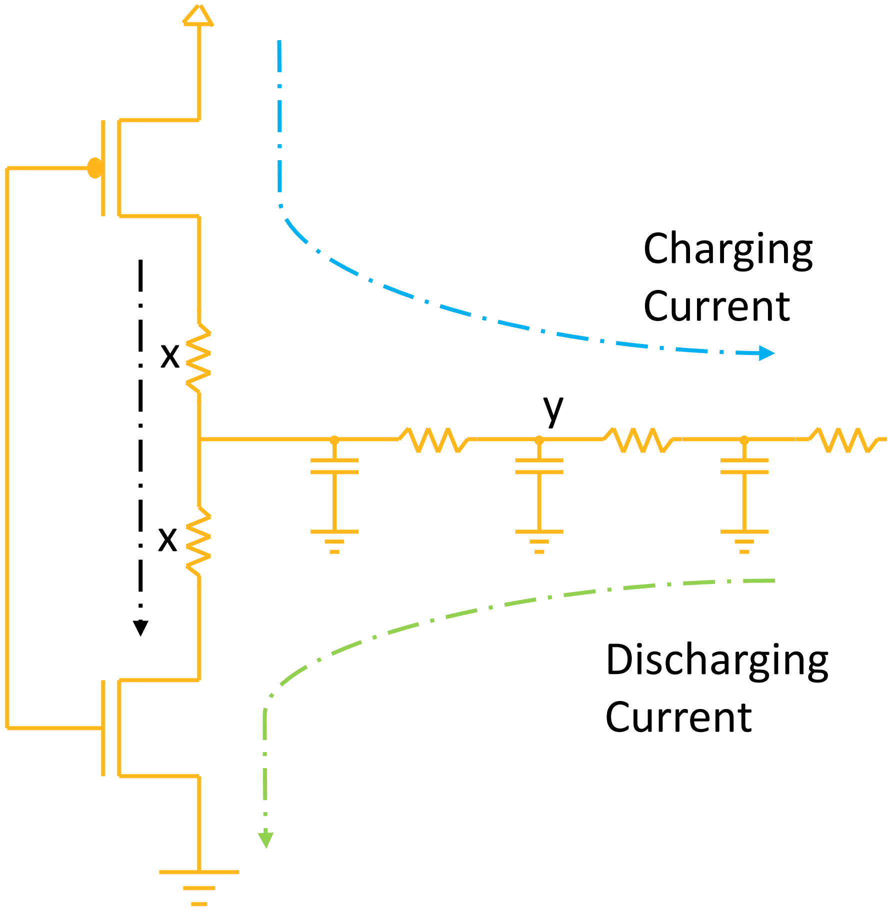
Power Electromigration

- Gradual displacement of ions in a Power interconnect
- Occurs when the current density is sufficiently high to cause the drift of metal ions in the direction of the electron flow.
- The intensity of the effect depends on two factors
  - Magnitude of forces which tend to **hold** the ions in place i.e
    - nature of the conductor,
    - crystal size,
    - interface and
    - grain-boundary chemistry.
  - Magnitude of forces which tend to **dislodge** the ions i.e,
    - current density,
    - temperature and
    - mechanical stress

Uni/Bi-directional Currents

- **Unidirectional**
  - Current through the x-segment is unidirectional
  - For rise/fall transition, the current flows in the same direction
  - Power/Ground rails have unidirectional current flow

- **Bidirectional**
  - Current through the y-segment is bidirectional
  - Forward current while charging/ Reverse while discharging
  - Average current is close to 0, due to recovery from discharge



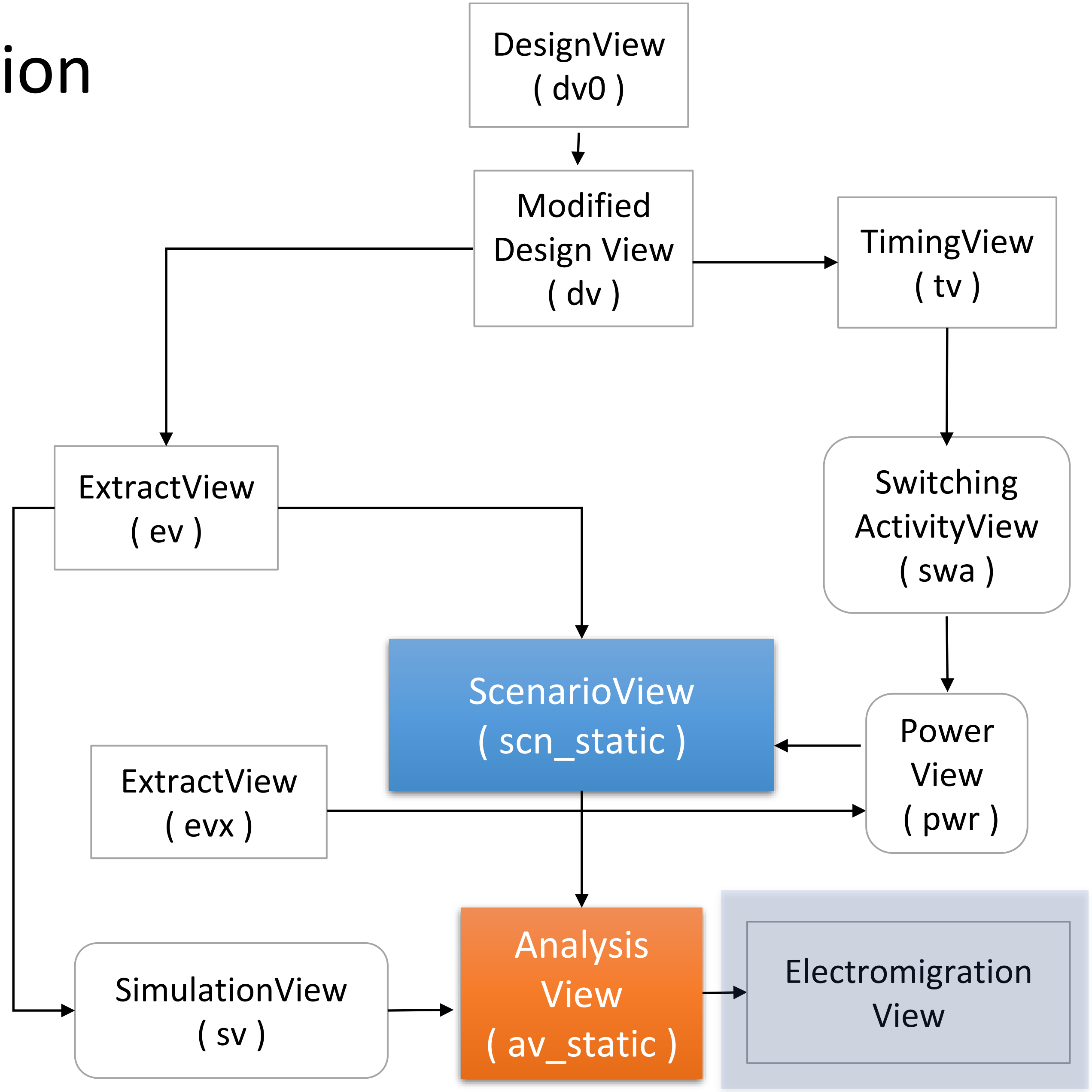
DC Power EM Analysis Flow

Performing Power Electromigration

ElectroMigrationView

- A specific view performing EM limit checks
- Can perform the following checks: Avg, RMS and Peak on PG nets
- Static AnalysisView has DC (Avg) currents
- For DC EM (Avg), feed in a Static AnalysisView
- For RMS or Peak, feed in a Transient AnalysisView
- EM rules come from the technology information stored in the Technology View
- For more details , see :

  help(SeaScapeDB.create_electro_migration_view)



Preparing for Power Electromigration run

File : scripts/args.py

```
pem_dc_args = dict(
    tag='pem_dc',
    mode='dc',
    temperature_em=125,
    options=options)
```

File : scripts/electromigration_dc.py

```
pem_dc = db.create_electromigration_view(av_static, **pem_dc_args)

reports_dir = 'reports/powerem_analysis'
gp_util.makedir(reports_dir)
emir_reports.write_em_metal_report(pem_dc, reports_dir+/'DC_EM_metal_report.rpt')
emir_reports.write_em_via_report(pem_dc, reports_dir+'/DC_EM_via_report.rpt')
```

Looking At EM Analysis Reports

EM Analysis Reports will Be Available In 'reports/powerem_analysis' directory

```
DC_EM_metal_report.rpt
DC_EM_via_report.rpt
```

em_type = DC, ignore_sliver = True, 'length' column is blech length

layer	from (u)	to (u)	length (u)	width (u)	current (A)	constraint (A)	violation (%)	status	net
metal1	(495.14,1136.8)	(495.23,1136.8)	393.6	0.17	0.0001134	0.0003156	35.94	PASS	VDD
metal1	(495.23,1136.8)	(495.785,1136.8)	393.6	0.17	0.0001133	0.0003156	35.9	PASS	VDD
metal1	(511.565,950.6)	(511.615,950.6)	393.6	0.17	0.0001111	0.0003156	35.2	PASS	VSS
metal1	(271.32,88.2)	(271.615,88.2)	406.5	0.17	0.0001096	0.0003156	34.73	PASS	VSS
metal1	(270.455,88.2)	(271.32,88.2)	406.5	0.17	0.0001096	0.0003156	34.73	PASS	VSS
metal1	(256.5,238.0)	(256.775,238.0)	407.4	0.17	0.0001067	0.0003156	33.82	PASS	VDD
metal1	(256.215,238.0)	(256.5,238.0)	407.4	0.17	0.0001067	0.0003156	33.82	PASS	VDD
metal1	(256.025,238.0)	(256.215,238.0)	407.4	0.17	0.0001067	0.0003156	33.82	PASS	VDD
metal1	(255.14,238.0)	(256.025,238.0)	407.4	0.17	0.0001067	0.0003156	33.82	PASS	VDD
metal1	(256.775,238.0)	(257.54,238.0)	407.4	0.17	0.0001065	0.0003156	33.76	PASS	VDD
metal1	(257.54,238.0)	(258.145,238.0)	407.4	0.17	0.0001064	0.0003156	33.71	PASS	VDD

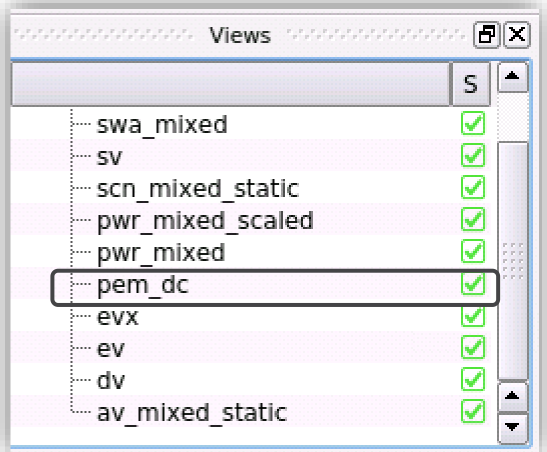
DC EM METAL REPORT

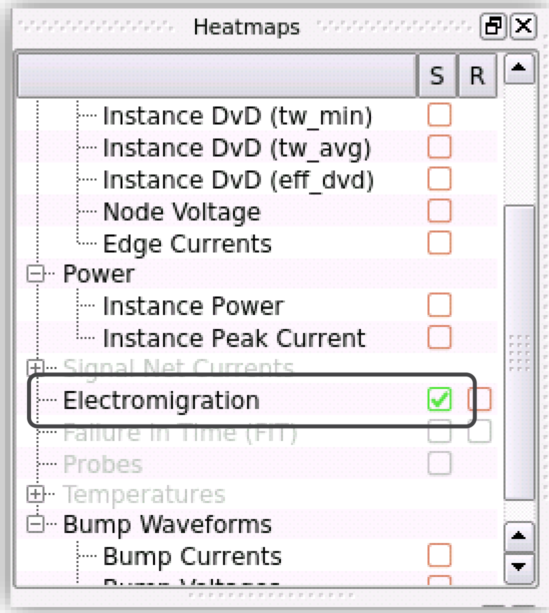
em_type = DC, ignore_sliver = True

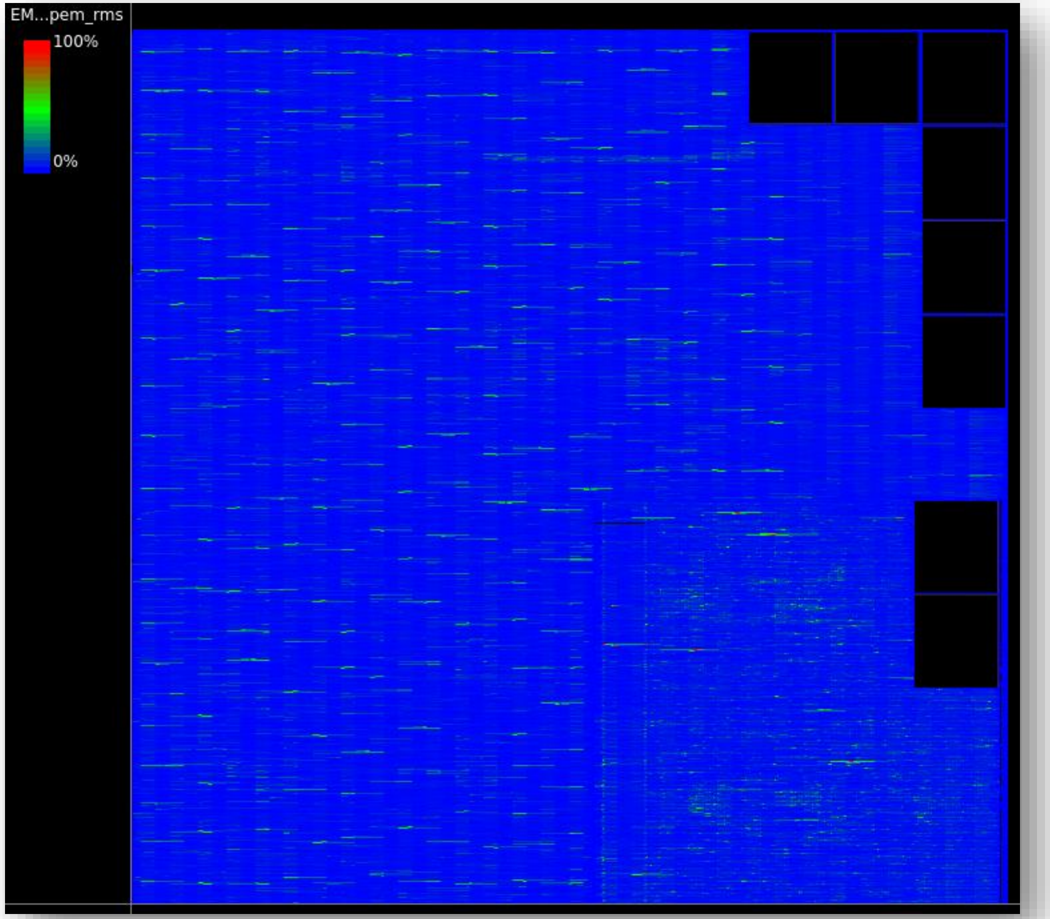
layer	loc_x (u)	loc_y (u)	via_length (u)	via_width (u)	current (A)	constraint (u)	violation (%)	status	net
via11	1088.4	61.6	0.8	0.8	0.0001266	0.0004978	25.43	PASS	VSS
via11	1088.4	153.6	0.8	0.8	0.0001237	0.0004978	24.86	PASS	VSS
via11	1129.6	204.4	0.8	0.8	0.0001212	0.0004978	24.34	PASS	VDD
via11	71.6	1078.4	0.8	0.8	0.0001189	0.0004978	23.88	PASS	VSS
via11	1129.6	107.6	0.8	0.8	0.0001183	0.0004978	23.77	PASS	VDD
via11	950.4	204.4	0.8	0.8	0.0001169	0.0004978	23.49	PASS	VDD
via11	71.6	61.6	0.8	0.8	0.0001145	0.0004978	23	PASS	VSS
via8	894.6	935.2	0.4	0.4	2.847e-05	0.0001244	22.88	PASS	VDD
via8	892.08	935.2	0.4	0.4	2.846e-05	0.0001244	22.87	PASS	VDD
via11	71.6	153.6	0.8	0.8	0.0001137	0.0004978	22.85	PASS	VSS

DC EM VIA REPORT

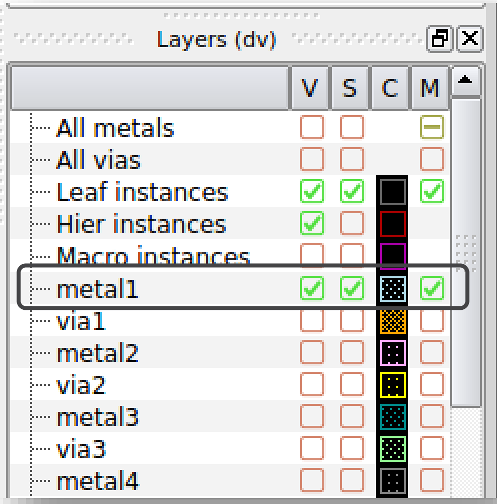
Examining EM Results In GUI



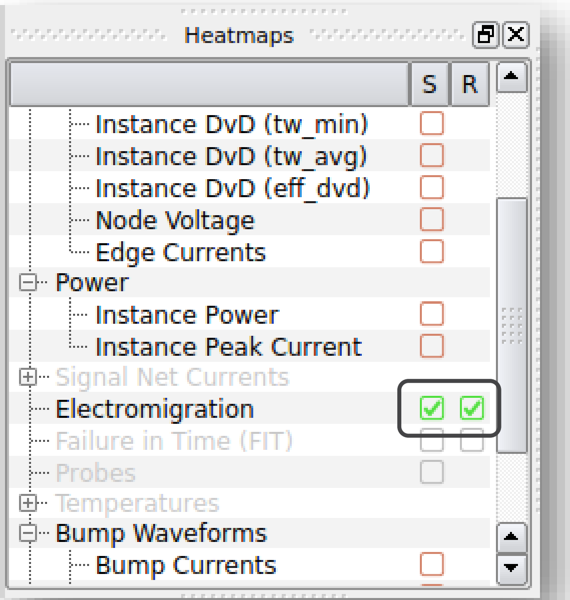


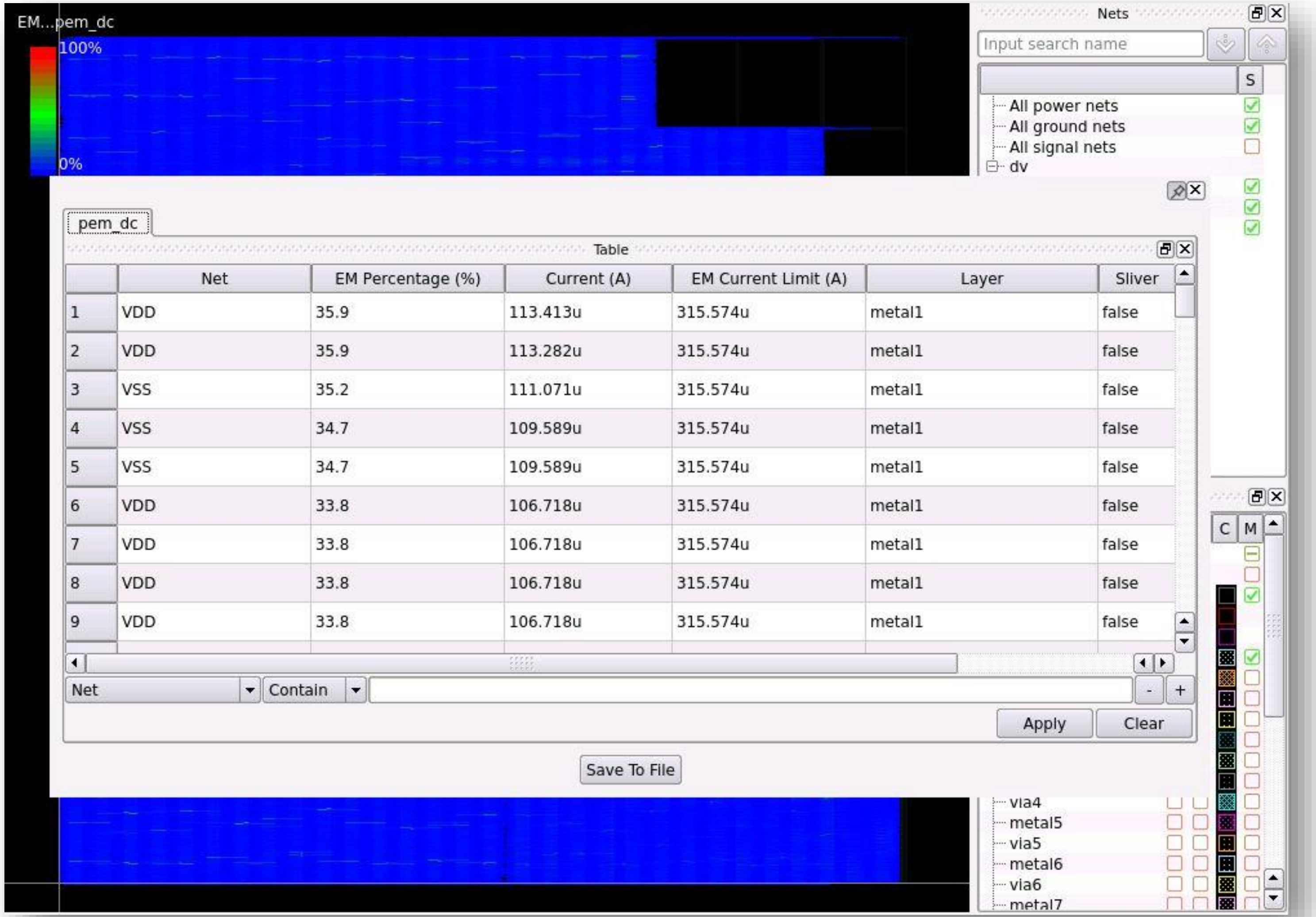


- ✓ Choose EM view
- ✓ Enable ‘Electromigration’ heatmap
- ✓ Choose the metal / via layer



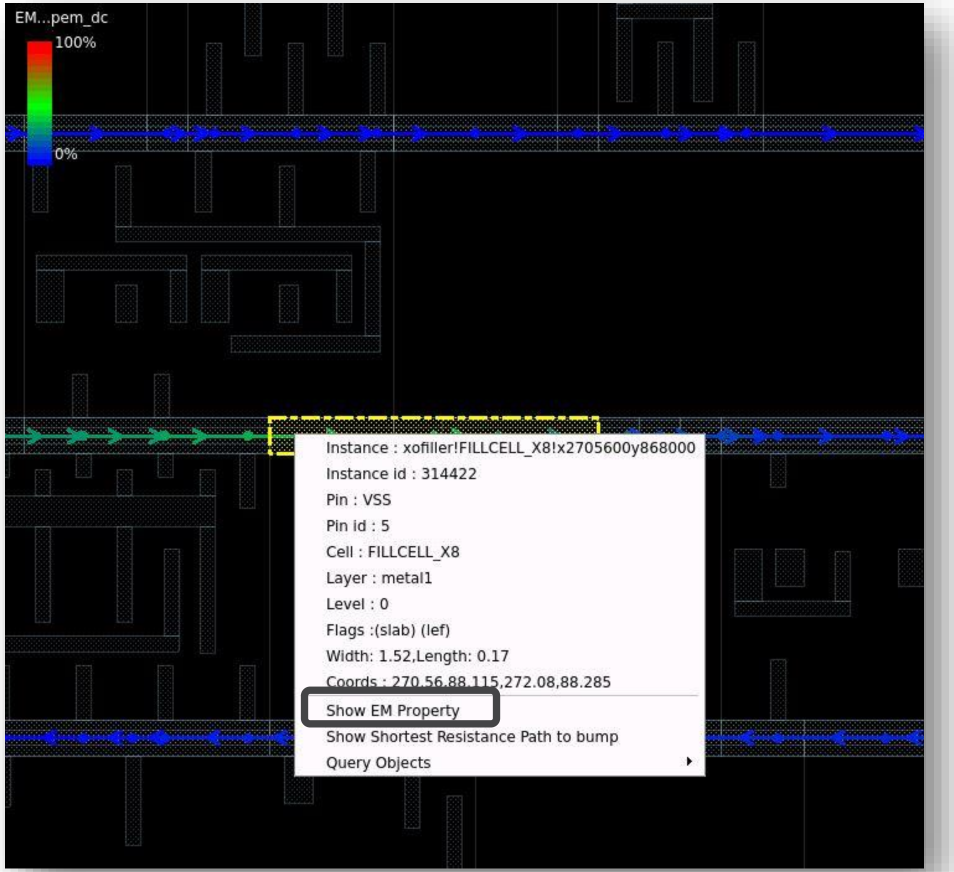
Examining EM Results In GUI

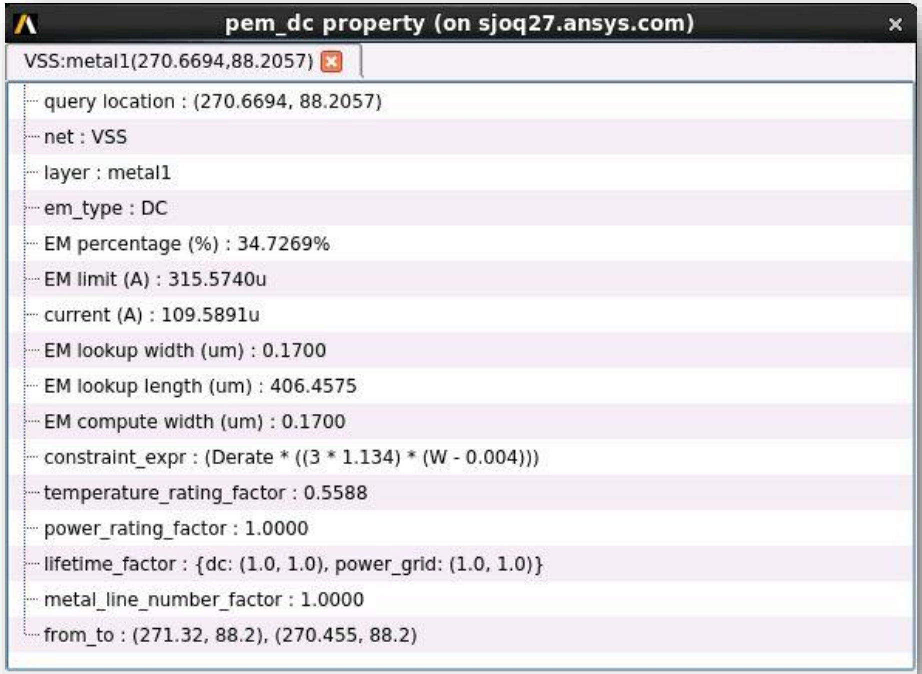




- ✓ Tick on box under letter "R" in Electromigration heatmap to open up EM violation browser
- ✓ Double click on the violation to zoom to the location in GUI

Examining EM Results In GUI





✓ Right click on a wire segment to see its EM property

Ansys

Thank You

Ansys

Appendix

Clock Definitions and Frequency in STA File

```
### STA Compact Version 1.2
######## Apache Design, Inc. A Subsidiary of ANSYS, Inc. ########
# File Type          : Timing Information (with signal load)
# Design Name        : Galaxy
# Date               : Sun Jan 19 10:37:22 2020
# ATE version        : 19.1.2   RHEL6 (Jun 14 00:53:34 2018)
# pt2timing version : 7.15
#################################################################


# bus_naming_style []
# hierarchy_separator /
# pin_delimiter /

#1) CLOCKS
#CLOCK <rise> <fall> <period> <root> <index>
CLOCK    0.000e-09    4.000e-09    8.000e-09 dco_clk 0
#TIMESCALE 1e-09
#CAPACITANCE_SCALE 1e-15
#RESISTANCE_SCALE 1e+00

#INSTANCE (376829)
#I $<id>/<instance_name>
#C <pin_name> <1|0>
#S <pin_name> <min_r_slew> <max_r_slew> <min_f_slew> <max_f_slew>
#T <pin_name> <is_clock> <min_r> <max_r> <min_f> <max_f> <clk_idx>
I ZBUF_2_inst_65483
S A 0.001 0.001 0.001 0.001
S Z 0.007 0.007 0.006 0.006
T Z 0 4.795 4.804 4.795 4.803 0
```

Freq and Clock Index defined here

Clock index for each instance defined in last column

Clock Network / Transition Time / TW in STA File

```
#INSTANCE (376829)
#I $<id>/<instance_name>
#C <pin_name> <1|0>
#S <pin_name> <min_r_slew> <max_r_slew> <min_f_slew> <max_f_slew>
#T <pin_name> <is_clock> <min_r> <max_r> <min_f> <max_f> <clk_idx>
I ZBUF_2_inst_65483
S A 0.001 0.001 0.001 0.001
S Z 0.007 0.007 0.006 0.006
T Z 0 4.795 4.804 4.795 4.803 0
I SGI3_256
S A 0.027 0.043 0.027 0.043
S ZN 0.009 0.009 0.006 0.006
L ZN 0.065959 170.962 1.554208 1 0.000000
C ZN 0
I HFSBUF_1833_9468
S A 0.073 0.074 0.070 0.071
S Z 0.083 0.083 0.079 0.079
L Z 3.438920 396.829 65.090363 1 44.845505
```

T : specifies the timing window information for an instance pin
3rd column in STA File TW section i.e <is_clock> tells whether the output of instance belongs to clock tree network

RISE/FALL transition time for both input and output pins
S : Specifies the transition time information for an instance pin
Transition time used in RedHawk-SC is as per slew_type Key value given while reading STA file.

SDC file example

Supported Commands

- set_units
- Create_clock
- set_driving_cell
- set_input_transition
- set_input_delay
- set_load
- set_case_analysis
- set_ideal_network
- set_ideal_latency
- set_ideal_transition
- set_clock_transition
- create_generated_clock
- set_sense
- set_annotated_transition

Setting units

Creating Clock

Setting load

```
# Setting Units
set_units -time ns -resistance kOhm -capacitance pF -voltage V -current mA

# Creating Clock
create_clock [get_ports dco_clk]  -name dco_clk  -period 8  -waveform {0 4}

# Setting Load on output ports
set_load -pin_load 0.0037518 [get_ports aclk_mcout]
set_load -pin_load 0.0037518 [get_ports aclk_en_mcout]
set_load -pin_load 0.0037518 [get_ports dbg_freeze_mcout]

# Setting input ports to 0/1
set_case_analysis 1 [get_ports reset_n]
set_case_analysis 0 [get_ports scan_mode]
set_case_analysis 0 [get_ports scan_enable]
```

Setting input port values

SAIF – Switching Activity Interchange Format

- SAIF : Switching Activity Interchange Format
- SAIF : Generated by 3rd party tools, contains switching activity for each instance in the design
- SAIF File Format:

```
// SAIF file for seq1_b2
(SAIFILE
  (DATE "Tue Jan 13 15:42:39 PST 2015")
  (SAIFVERSION "2.0")
  (DIRECTION "backward")
  (TIMESCALE 1 ns)
  (DURATION 100)
  (INSTANCE U3
    (INSTANCE FF1
      (NET
        (Q (T0 1) (T1 99) (TC 200))
      )
    )
    (INSTANCE FF2
      (NET
        (Q (T0 2) (T1 98) (TC 200))
      )
    )
  )
  (INSTANCE U4
    (INSTANCE FF1
      (NET
        (Q (T0 3) (T1 97) (TC 200))
      )
    )
    (INSTANCE FF2
      (NET
        (Q (T0 4) (T1 96) (TC 200))
      )
    )
  )
)
```

- T0: Total time design object is in “0” state
- T1: Total time design object is in “1” state
- TX: Total time design object is in “X” state
- TZ: Total time design object is in “Z” state

The toggle attributes can be summarized as :

- TC : Number of “0” to “1” and “1” to “0” transitions

Switching Activity View Arguments

Argument	Description
propagation_style	Type of propagation for instance pins without data (type=str, default_value='propagate_activity', constraint="One of ['propagate_activity', 'assign_default_activity', 'no_propagation']")
object_settings	Dict of settings for various scopes (type=dict, default_value={}, constraint="dict of design, cell and/or instance values")
settings	Activity settings (type=dict, default_value=None, constraint="dict of settings")
use_sta_clock_info	Use clock info from STA file if available, otherwise use default clock info (type=bool, default_value=False)
infer_icg_enable_signals_from_fanin_prop	Infer ICG behavior from signals propagated to enable pin (type=bool, default_value=False)
disable_clocks	List of clocks to disable (type=list, default_value=None, constraint="list of clock names or patterns to disable")
clock_source_toggle_rates	Dict of {clock_name : toggle_rate} to override default toggle rate of 2.0 on specific clocks (type=dict, default_value=None, constraint="dict")
options	View creation options, typically from get_default_options() (type=object, default_value=None)
tag	Name of the view (type=str, default_value='swa')

Querying Sources of Activity in SwitchingActivityView

Source	Description
Propagation	Activity is annotated from Switching Activity View Propagation
DefaultActivity	Activity is annotated from default toggle rates
InferIcg	When infer_icg_enable_signals_from_fanin_prop is True in create_switching_activity_view, some instances will have this source.
ConstFromPropagation	It is constant but from propagation. For example, if a buffer's input is const 0, then we know the output is const 0 from propagation.
ConstFromLso	Const from LSO (Logic Signal Override)
ConstFromSca	Const from Set Case Analysis in SDC
ConstFromTiming	Activity is annotated from CONST in STA
SaifOrVcd	Activity is annotated from SAIF/VCD. Pins are connected to PG Net
ConstFromLib	Constant from Library

Sources of Activity in SwitchingActivityView .. Continued

Source	Description
ConstFromDisconnection	Constant for floating input pins
ConstFromPGNet	Constant from PG connection
PinLevelSettings	Settings from leaf_instance_pin_values

Internal Power Calculation from Liberty file

```
pin(Y) {
  direction : output ;
  function : "(A&B)" ;
  max_capacitance : 0.0393506 ;
  max_transition : 0.388 ;
  min_capacitance : 0.0001 ;
  output_voltage : default ;
  related_ground_pin : VSS ;
  related_power_pin : VDD ;
  power_down_function : "!VDD + VSS" ;

  internal_power() {
    related_pin : "A" ;
    when : "B" ;

    fall_power(pwr_tin_oload_7x7) {
      index_1("0.001558, 0.00850942, 0.0329104, 0.0795077, \
              0.152104, 0.253965, 0.388");
      index_2("0.0001, 0.00080605, 0.00328444, 0.00801729, \
              0.0153908, 0.0257368, 0.0393506");
      values("0.00136541, 0.00138817, 0.00141447, 0.00142179, 0.00142527, 0.00142405, 0.00141913",\
             "0.00135146, 0.0013686, 0.00139995, 0.00140762, 0.00141021, 0.00140938, 0.00140759",\
             "0.00135605, 0.00136041, 0.00138305, 0.0013966, 0.0014029, 0.00140353, 0.00139994",\
             "0.0014737, 0.00145205, 0.00144857, 0.00145818, 0.00146426, 0.00146623, 0.00146669",\
             "0.00169968, 0.00165253, 0.00162487, 0.00162144, 0.00162282, 0.00162447, 0.00162537",\
             "0.00207387, 0.00199098, 0.00192903, 0.00190567, 0.00189653, 0.00189581, 0.00189495",\
             "0.00260845, 0.00249534, 0.00237953, 0.00232992, 0.00230902, 0.0022979, 0.00229404");
    }

    rise_power(pwr_tin_oload_7x7) {
      index_1("0.001558, 0.00850942, 0.0329104, 0.0795077, \
              0.152104, 0.253965, 0.388");
      index_2("0.0001, 0.00080605, 0.00328444, 0.00801729, \
              0.0153908, 0.0257368, 0.0393506");
      values("0.000801811, 0.000786919, 0.000776793, 0.000720899, 0.0006235, 0.00049138, 0.000329456",\
             "0.000781487, 0.000777169, 0.000739173, 0.000664675, 0.000640959, 0.000373424, 0.000145712",\
             "0.00078938, 0.00075972, 0.000729628, 0.000653343, 0.000625186, 0.000357625, 0.000388812",\
             "0.000914398, 0.000881621, 0.000827515, 0.000721483, 0.000596325, 0.000436321, 0.00018914",\
             "0.00116296, 0.00110022, 0.00107452, 0.000951866, 0.00076779, 0.000752509, 0.000605146",\
             "0.00156306, 0.00146795, 0.0013935, 0.00138966, 0.00114837, 0.00102613, 0.000880344",\
             "0.00212187, 0.00198954, 0.00186016, 0.00183001, 0.00180936, 0.00153084, 0.0013616");
    }
  }
```

ARC : A->Y

A->Y Fall

A->Y Rise

A->Y Power Tables for RISE and FALL (B=1)

```
internal_power() {
  related_pin : "B" ;
  when : "A" ;

  fall_power(pwr_tin_oload_7x7) {
    index_1("0.001558, 0.00850942, 0.0329104, 0.0795077, \
            0.152104, 0.253965, 0.388");
    index_2("0.0001, 0.00080605, 0.00328444, 0.00801729, \
            0.0153908, 0.0257368, 0.0393506");
    values("0.00156223, 0.00157385, 0.00159369, 0.00159842, 0.00159772, 0.00159544, 0.00159448",\
           "0.001542, 0.00155447, 0.00157598, 0.00157887, 0.00157896, 0.00157691, 0.00157752",\
           "0.00154507, 0.00154641, 0.00156356, 0.00157234, 0.00157455, 0.00157253, 0.00157139",\
           "0.00165967, 0.00163874, 0.0016298, 0.0016357, 0.00163968, 0.00163966, 0.00164008",\
           "0.00188399, 0.00183691, 0.00180792, 0.00179971, 0.00180036, 0.00179874, 0.00179752",\
           "0.00223745, 0.00216153, 0.00209938, 0.00207581, 0.00206672, 0.00206187, 0.00205918",\
           "0.00273264, 0.00262594, 0.00251762, 0.00246981, 0.00244817, 0.0024356, 0.00242852");
  }

  rise_power(pwr_tin_oload_7x7) {
    index_1("0.001558, 0.00850942, 0.0329104, 0.0795077, \
            0.152104, 0.253965, 0.388");
    index_2("0.0001, 0.00080605, 0.00328444, 0.00801729, \
            0.0153908, 0.0257368, 0.0393506");
    values("0.000789886, 0.000773287, 0.000751985, 0.000629807, 0.000513227, 0.000470961, 0.000304004",\
           "0.000762475, 0.000748451, 0.000721467, 0.000693536, 0.000550965, 0.000384068, 0.000186722",\
           "0.000745391, 0.000720881, 0.00069[illegible]266, 0.000623768, 0.000491282, 0.000328949, 0.000335019",\
           "0.000816587, 0.000795403, 0.000731141, 0.000649571, 0.00052154, 0.000366326, 0.000110363",\
           "0.00100412, 0.000957324, 0.000943806, 0.000836508, 0.000756333, 0.000641852, 0.000261879",\
           "0.00133857, 0.00126837, 0.00120951, 0.00122149, 0.00101886, 0.00092854, 0.000766882",\
           "0.00182161, 0.00171434, 0.00161243, 0.00159722, 0.00161677, 0.00123554, 0.00120427");
  }
}
```

ARC : B->Y

B->Y Fall

B->Y Rise

B->Y Power Tables for RISE and FALL (A=1)

Internal Power Calculation from Liberty file

```
pin(A) {
  capacitance : 0.000569394 ;
  direction : input ;
  fall_capacitance : 0.000575371 ;
  input_voltage : default ;
  max_transition : 0.388 ;
  related_ground_pin : VSS ;
  related_power_pin : VDD ;
  rise_capacitance : 0.000563417 ;

  internal_power() {
    when : "!B" ;

    fall_power(pwr_tin_7) {
      index_1("0.001558, 0.00850942, 0.0329104, 0.0795077, \
               0.152104, 0.253965, 0.388");
      values("0.000255115, 0.000258959, 0.000262106, 0.000262646, \
             0.000262884, 0.000263084, 0.000263268");
    }

    rise_power(pwr_tin_7) {
      index_1("0.001558, 0.00850942, 0.0329104, 0.0795077, \
               0.152104, 0.253965, 0.388");
      values("-0.000184331, -0.000188318, -0.000189967, \
             -0.000190232, -0.00019009, -0.000189947, -0.000190011");
    }
  }
}
```

A FALL, when B=0

A RISE, when B=0

```
pin(B) {
  capacitance : 0.000578491 ;
  direction : input ;
  fall_capacitance : 0.000572989 ;
  input_voltage : default ;
  max_transition : 0.388 ;
  related_ground_pin : VSS ;
  related_power_pin : VDD ;
  rise_capacitance : 0.000583994 ;

  internal_power() {
    when : "!A" ;

    fall_power(pwr_tin_7) {
      index_1("0.001558, 0.00850942, 0.0329104, 0.0795077, \
               0.152104, 0.253965, 0.388");
      values("0.0002912, 0.000280596, 0.000276642, 0.00027454, \
             0.00027339, 0.000272839, 0.000272726");
    }

    rise_power(pwr_tin_7) {
      index_1("0.001558, 0.00850942, 0.0329104, 0.0795077, \
               0.152104, 0.253965, 0.388");
      values("-0.000248317, -0.000251213, -0.000253997, \
             -0.000255192, -0.000255633, -0.000255783, -0.00025606");
    }
  }
}
```

B FALL, when A=0

B RISE, when A=0

A RISE/FALL Power tables when B=0

B RISE/FALL Power tables when A=0

Sources of Power in Power View Attributes

Source	Description
SwitchingActivity	Calculated from SwitchingActivityView toggle rates
IPF	Instance power file
PowerScaling	Scaled power in create_scaled_power_view
ModeControl	Calculated from switching activity but it has mode control
Scenario	When Scenario is given as input to PowerView

CellType Definition in Power Calculation Reports

Cell-type	Recognition Method
Combinational	Cells with no CLK pin in LIB file
Latch_and_FF	Cells with attributes ff, ff_bank, latch, latch_bank in .lib
Memory	memory attribute in LIB file
Clocked	Cells having Clock pin, but not classified as FF, Latch or Memory
I/O	pad_cell attribute in LIB file
decap	Is_decap_cell, is_filler_cell attribute in LIB file