RedHawk-SC Explorer & Reporting Utilities

RedHawk-SC Modular Training Series

Version : RedHawk-SC 2020_R3

Ansys

Prerequisite for the Training

No.	Training Program	Must be familiar with
1.	RedHawk-SC Modular Training	Creating Views including DesignView, TimingView and ExtractView
2.	Basic Python	Familiarity with Basic Python Programming. Ability to comprehend Python Scripts.

©2020 ANSYS, Inc. / Confidential

Table of Contents

- A few helpful and needed Python Functions
- Data Integrity Reports
- Grid Robustness Reporting
- Power Reporting
- Voltage Reporting

©2020 ANSYS, Inc. / Confidential

A few helpful Python Constructs

functools.partial

- The Python Partial Function
  - Convert a ‘n’ argument function to a ‘m’ argument function (m < n)

```
In [1]: from functools import partial

In [2]: def add(a, b):  return a+b

In [3]: add(2, 3)
Out[3]: 5

In [4]: add2 = partial(add, b=2)

In [5]: add2(6)
Out[5]: 8

In [6]: add5 = partial(add, b=5)

In [7]: add5(2)
Out[7]: 7

In [8]: add2(3) + add5(5)
Out[8]: 15
```

RedHawk-SC automatically imports partial to its environment when running

©2020 ANSYS, Inc. / Confidential

str.format

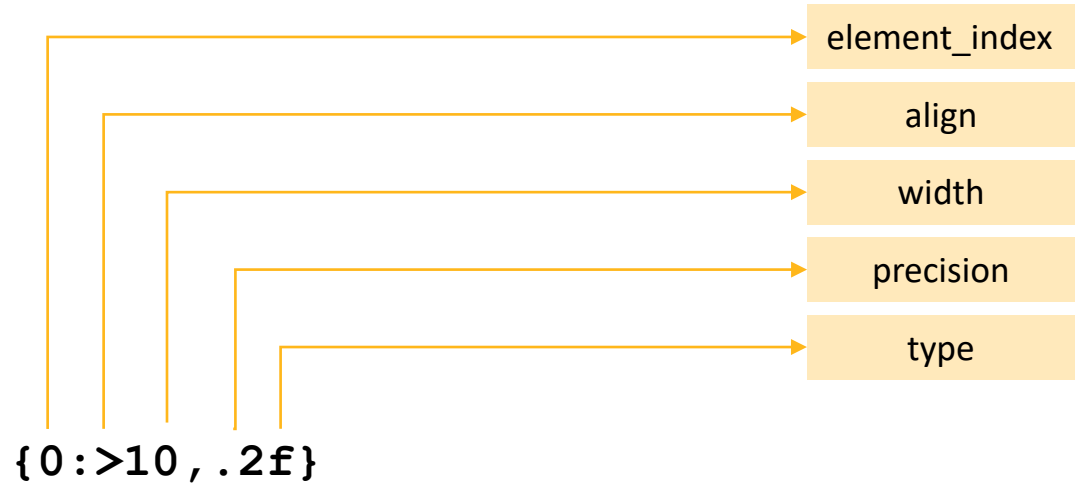
```
string = '{0:10.5f} {1:25} {2}'.format(234.212345, 'Hello', 'Hero')
print(string + '\n')
# >>> _234.21234_Hello_____________________Hero
```


©2020 ANSYS, Inc. / Confidential

str.format

Simplified Format language specification*

```
{element_index:format_spec}
element_index := integer
format_spec := [[align][width][,][.precision][type]]
align := '<', '>', '^'
width := integer
precision := integer
type := 'f', 'g', '%'
```



* Refer to https://docs.python.org/2/library/string.html#format-string-syntax for the full syntax spec

©2020 ANSYS, Inc. / Confidential

gp_delayed_object

- SeaScape Specific object to Schedule Jobs/Functions on the Workers
- Allows fall-through
  - Does not hold up the Master Console when executing
- Call `get` on the `gp_delayed_object` to use the results

- Please refer to the next few slides for details

©2020 ANSYS, Inc. / Confidential

Serial Execution of Jobs

```
def long_calculation(args):
    return do_something(args)

def longer_calculation(args):
    return do_something_else(args)

def process_result(aa, bb):
    return calculate(aa, bb)

def write_to_file(file_name, value):
    with open(file_name, 'w') as fp:
        fp.write(value + '\n')


intermediate_result_1 = long_calculation(args)
intermediate_result_2 = longer_calculation(args_2)
final_result = process_result(intermediate_result_1, intermediate_result_2)
write_to_file('output', final_result)
print("Calculated")
```


©2020 ANSYS, Inc. / Confidential

Making the Calculations fall-through with `gp_delayed_object`

- We provide a decorator, `sch_func` that can be applied to a function to make it a `gp_delayed_object`
- This allows the function to run on workers, freeing up the Master Console and making the whole script fall-through

```
def long_calculation(args):
    return calculate(args)

d_result = gp_delayed_object(partial(long_calculation, args=args))
```

```
@sch_func
def long_calculation(args):
    return calculate(args)

d_result = long_calculation(args)
```


©2020 ANSYS, Inc. / Confidential

Making the Calculations fall-through with `gp_delayed_object`

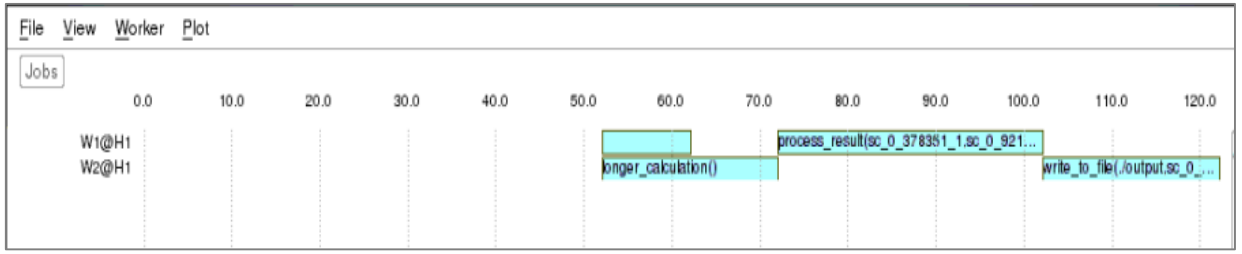
```
@sch_func
def long_calculation(args):
    return do_something(args)

@sch_func
def longer_calculation(args):
    return do_something_else(args)

@sch_func
def process_result(aa, bb):
    data_1 = aa.get()
    data_2 = bb.get()
    return calculate(data_1, data_2)

@sch_func
def write_to_file(file_name, value):
    with open(file_name, 'w') as fp:
        fp.write('{0}\n'.format(value.get()))

intermediate_result_1 = long_calculation(args)
intermediate_result_2 = longer_calculation(args_2)
final_result = process_result(intermediate_result_1, intermediate_result_2)
write_to_file('output', final_result)
print("Submitted")
```



©2020 ANSYS, Inc. / Confidential

Data Integrity Reports

Data Integrity Reports: Design View

`data_integrity_reports.create_dv_di_reports`

```
di = data_integrity_reports.create_dv_di_reports(dv)
data_integrity_summary = di.get()
```

```
{'apl_cap_check': {'Combinational': {'covered': 67, 'total': 67},
                   'Decap': {'covered': 0, 'total': 0},
                   'Filler': {'covered': 0, 'total': 0},
                   'ICG': {'covered': 0, 'total': 0},
                   'MBFF': {'covered': 0, 'total': 0},
                   'Macro': {'covered': 1, 'total': 1},
                   'Power_Gate': {'covered': 0, 'total': 1},
                   'Sequential': {'covered': 8, 'total': 8}},
 'apl_current_check': {'Combinational': {'covered': 67, 'total': 67},
                       'Decap': {'covered': 0, 'total': 0},
                       'Filler': {'covered': 0, 'total': 0},
                       'ICG': {'covered': 0, 'total': 0},
                       'MBFF': {'covered': 0, 'total': 0},
                       'Macro': {'covered': 1, 'total': 1},
                       'Power_Gate': {'covered': 0, 'total': 1},
                       'Sequential': {'covered': 8, 'total': 8}},
```


©2020 ANSYS, Inc. / Confidential

Design View Data Integrity Reports: Available Checks

`data_integrity_reports.create_dv_di_reports`

Available Design View Data Integrity Checks	
`apl_cap_check`	List of cells that have no APL Cap Models in the design
`apl_current_check`	List of cells that have no APL Current Models in the design
`ccs_cap_check`	List of Cells that have no CCS Power Cap Models in the Design
`ccs_current_check`	List of Cells that have no CCS Current Models in the Design
`ccs_timing_check`	List of Cells that have no CCS Timing Models in the Design
`lef_check`	List of Cells that do not have a LEF Cell in the Design
`lib_check`	List of Cells that do not have a Liberty Model in the Design
`lib_nldm`	List of Cells that do not have Liberty Delay Models in the Design
`lib_nlpm`	List of Cells that do not have Liberty Power Models in the Design
`macro_model_check`	List of Macro Cells that do not have Macro Models in the Design

©2020 ANSYS, Inc. / Confidential

Design View Data Integrity Reports: Argument(s)

`data_integrity_reports.create_dv_di_reports`

Required Argument
dv

Optional Arguments	
celltype_excludes	By default, Filler and Decap will be excluded from all checks
detailed_reports	False
detailed_reports_directory	“./data_integrity_reports”

©2020 ANSYS, Inc. / Confidential

Design View Data Integrity Reports: celltype_excludes

data_integrity_reports.create_dv_di_reports

Valid options for celltype_excludes
Combinational
Decap
Filler
ICG
MBFF
Macro
Power Gate
Sequential

©2020 ANSYS, Inc. / Confidential

DesignView Data Integrity Reports: Usage Examples

Create the DesignView data integrity reports for the design

```
data_integrity_reports.create_dv_di_reports(dv, detailed_reports=True)
```


©2020 ANSYS, Inc. / Confidential

DesignView Data Integrity Reports: Usage Examples

Using celltype_excludes to control the type of cells being reported

```
exclude_celltypes = { 'apl_cap_check' : ['Filler', 'Decap'],
                      'ccs_timing_check': [],
                      'lef_check' : ['Combinational'] }

data_integrity_reports.create_dv_di_reports(dv, detailed_reports=True,
                                            celltype_excludes=exclude_celltypes)
```


©2020 ANSYS, Inc. / Confidential

SPEF Data Integrity Reports

`data_integrity_reports.create_spef_data_integrity_reports`

```
sdi = data_integrity_reports.create_spef_data_integrity_reports(evx)
spef_data_integrity_summary = sdi.get()
```

```
spef_data_integrity_summary

{'di_data': <gp.ChunkedData object at 0x2b8009f2eb90>,
'summary': {'spef_check': {'Combinational': {'uncovered_instances': 9890},
                           'Sequential': {'uncovered_instances': 70736},
                           'total': {'total_instances': 380028, 'uncovered_instances': 80626}},
            'zero_cap_check': {'total': {'total_instances': 380028}}}}
>>>
```


©2020 ANSYS, Inc. / Confidential

SPEF Data Integrity Reports: Available Checks

```
data_integrity_reports.create_spef_data_integrity_reports
```

Available SPEF Data Integrity Reports	
spef_check	List of Instances that have all their pins covered in the input SPEF
zero_cap_check	List of Instances that have non-zero capacitance annotated from the input SPEF

©2020 ANSYS, Inc. / Confidential

SPEF Data Integrity Reports: Argument(s)

`data_integrity_reports.create_spef_data_integrity_reports`

Required Argument
evx

Optional Arguments	
scenario_view	None
detailed_reports	False
filter_func	None
filter_types	
detailed_reports_file_name	“./spef_detailed_reports.xxx”

©2020 ANSYS, Inc. / Confidential

SPEF Data Integrity Reports: `filter_types`

```
data_integrity_reports.create_spef_data_integrity_reports
```

Valid options for `filter_types`
`Combinational`
`Decap`
`Filler`
`ICG`
`MBFF`
`Macro`
`Power Gate`
`Sequential`

©2020 ANSYS, Inc. / Confidential

SPEF Data Integrity Reports: Usage Examples

Create the detailed SPEF Data Integrity Reports for the design

```
data_integrity_reports.create_spef_data_integrity_reports(evx, detailed_reports=True)
```

#Instance	Pin	Net
ZBUF_314_inst_65462	Z	ZBUF_314_4214
ZBUF_3658_inst_51786	Z	ZBUF_3658_69
ZBUF_603_inst_65321	Z	ZBUF_603_4207
ZBUF_2_inst_54949	Z	ZBUF_2_2382
core2.regfile_data_memory.DFF_X1_100	QN	core2.regfile_data_memory._26781_
core2.regfile_data_memory.DFF_X1_101	QN	core2.regfile_data_memory._26782_
core2.regfile_data_memory.DFF_X1_102	QN	core2.regfile_data_memory._26783_
core2.regfile_data_memory.DFF_X1_103	QN	core2.regfile_data_memory._26784_
core2.regfile_data_memory.DFF_X1_104	QN	core2.regfile_data_memory._26785_
core2.regfile_data_memory.DFF_X1_1042	QN	core2.regfile_data_memory._27723_
core2.regfile_data_memory.DFF_X1_1043	QN	core2.regfile_data_memory._27724_
core2.regfile_data_memory.DFF_X1_105	QN	core2.regfile_data_memory._26786_

spef_check

#Instance	Pin	Net
ZBUF_314_inst_65462	Z	ZBUF_314_4214
ZBUF_3658_inst_51786	Z	ZBUF_3658_69
ZBUF_603_inst_65321	Z	ZBUF_603_4207
ZBUF_2_inst_54949	Z	ZBUF_2_2382
core2.regfile_data_memory.DFF_X1_100	QN	core2.regfile_data_memory._26781_
core2.regfile_data_memory.DFF_X1_101	QN	core2.regfile_data_memory._26782_
core2.regfile_data_memory.DFF_X1_102	QN	core2.regfile_data_memory._26783_
core2.regfile_data_memory.DFF_X1_103	QN	core2.regfile_data_memory._26784_
core2.regfile_data_memory.DFF_X1_104	QN	core2.regfile_data_memory._26785_
core2.regfile_data_memory.DFF_X1_1042	QN	core2.regfile_data_memory._27723_
core2.regfile_data_memory.DFF_X1_1043	QN	core2.regfile_data_memory._27724_
core2.regfile_data_memory.DFF_X1_105	QN	core2.regfile_data_memory._26786_

zero_cap_check

©2020 ANSYS, Inc. / Confidential

SPEF Data Integrity Reports: Usage Examples

Using `filter_types` to control the type of cells being reported

```
filter_types = ['Combinational', 'Filler', 'Decap']

data_integrity_reports.create_spef_data_integrity_reports(evx,
                        detailed_reports=True, filter_types=filter_types)
```


©2020 ANSYS, Inc. / Confidential

SPEF Data Integrity Reports: Usage Examples

Using `filter_func` to filter instances

```
def filter_func(instance, dv, skip_pattern):
    instance_name = dv.convert_to_name(instance)
    if re.search(skip_pattern, instance_name.get_name()):
        return True
    else:
        return False

data_integrity_reports.create_spef_data_integrity_reports(evx,
    detailed_reports=True, filter_func=partial(filter_func, dv=dv,
    skip_pattern=re.compile('.*filler.*', re.I))
```


©2020 ANSYS, Inc. / Confidential

STA Data Integrity Reports

data_integrity_reports.create_timing_view_data_integrity_reports

```
summary = data_integrity_reports.create_timing_view_data_integrity_reports(tv)
```

```
summary
{'chdata': None,
'slew_summary': defaultdict(<type 'int'>, {'Sequential': 70726, 'Macro': 8, 'Combinational': 283055}),
'total_count': defaultdict(<type 'int'>, {'Sequential': 72072, 'Macro': 8, 'Combinational': 305402}),
'tw_summary': defaultdict(<type 'int'>, {'Sequential': 70649, 'Macro': 8, 'Combinational': 290388})}
```


©2020 ANSYS, Inc. / Confidential

STA Data Integrity Reports: Available Checks

```
data_integrity_reports.create_timing_view_data_integrity_reports
```

Available STA Data Integrity Reports	
`tw_check`	List of instances which has no timing window for at least one of its pins
`slew_check`	List of instances which have no slew for at least one of its pins

©2020 ANSYS, Inc. / Confidential

STA Data Integrity Reports: Argument(s)

```
data_integrity_reports.create_timing_view_data_integrity_reports
```

Required Argument
tv

Optional Arguments	
per_instance_report	False
file_name	{‘slew’:’./inst_no_tw.rpt’, ‘timing_window’:’./inst_no_slew.rpt’}
exclude_cell_types	[‘Filler’, ‘Power Gate’, ‘Decap’]

©2020 ANSYS, Inc. / Confidential

STA Data Integrity Reports: exclude_cell_types

```
data_integrity_reports.create_timing_view_data_integrity_reports
```

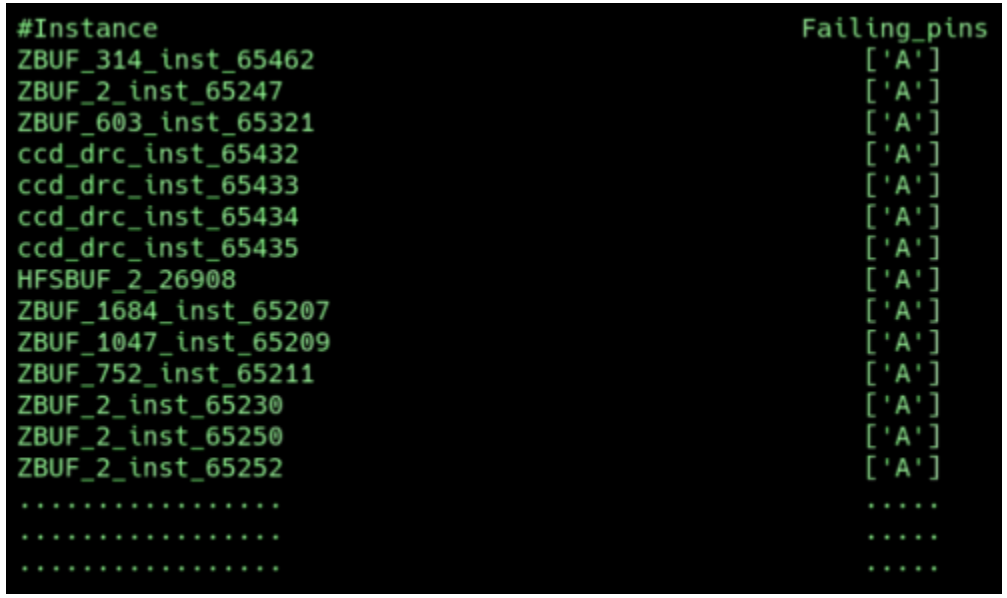
Valid options for **exclude_cell_types**
Combinational
Decap
Filler
ICG
MBFF
Macro
Power Gate
Sequential

©2020 ANSYS, Inc. / Confidential

STA Data Integrity Reports: Usage Examples

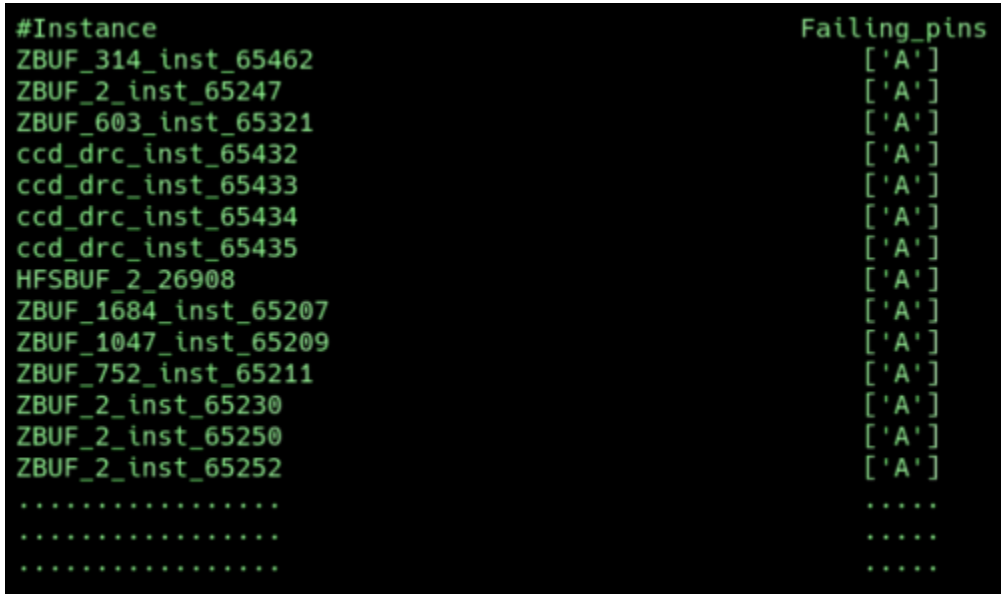
Create the detailed STA Data Integrity Reports for the design

```
data_integrity_reports.create_timing_view_data_integrity_reports(tv,
                                                                 per_instance_report=True)
```



#Instance	Failing_pins
ZBUF_314_inst_65462	['A']
ZBUF_2_inst_65247	['A']
ZBUF_603_inst_65321	['A']
ccd_drc_inst_65432	['A']
ccd_drc_inst_65433	['A']
ccd_drc_inst_65434	['A']
ccd_drc_inst_65435	['A']
HFSBUF_2_26908	['A']
ZBUF_1684_inst_65207	['A']
ZBUF_1047_inst_65209	['A']
ZBUF_752_inst_65211	['A']
ZBUF_2_inst_65230	['A']
ZBUF_2_inst_65250	['A']
ZBUF_2_inst_65252	['A']
.....	.....
.....	.....
.....	.....

sta_check



#Instance	Failing_pins
ZBUF_314_inst_65462	['A']
ZBUF_2_inst_65247	['A']
ZBUF_603_inst_65321	['A']
ccd_drc_inst_65432	['A']
ccd_drc_inst_65433	['A']
ccd_drc_inst_65434	['A']
ccd_drc_inst_65435	['A']
HFSBUF_2_26908	['A']
ZBUF_1684_inst_65207	['A']
ZBUF_1047_inst_65209	['A']
ZBUF_752_inst_65211	['A']
ZBUF_2_inst_65230	['A']
ZBUF_2_inst_65250	['A']
ZBUF_2_inst_65252	['A']
.....	.....
.....	.....
.....	.....

slew_check

©2020 ANSYS, Inc. / Confidential

STA Data Integrity Reports: Usage Examples

Using `exclude_cell_types` to control the type of cells being reported

```
exclude_cell_types = ['Combinational', 'Filler', 'Decap']

data_integrity_reports.create_timing_view_data_integrity_reports(evx,
      per_instance_report=True, exclude_cell_types=exclude_cell_types)
```


©2020 ANSYS, Inc. / Confidential

STA Data Integrity Reports: Usage Examples

Using `file_name` to change the output files

```
 file_name = { 'timing_view':'./no_tw.rpt',
               'slew'       :'./no_slew.rpt' }

data_integrity_reports.create_timing_view_data_integrity_reports(tv,
                          per_instance_report=True, file_name=file_name)
```


©2020 ANSYS, Inc. / Confidential

Construction Check Reports: Disconnected Instances

`data_integrity_reports.write_unconnected_pg_pins_report`

```
data_integrity_reports.write_unconnected_pg_pins_report(ev)
```

# loc_x	loc_y	cell_name	pin	net	logical_disconnect	physical_disconnect	instance
1215.62	222.6	INV_X4	VDD	VDD	0	1	ZINV_133_inst_52478
1215.62	222.6	INV_X4	VSS	VSS	0	1	ZINV_133_inst_52478
1216.57	222.6	INV_X4	VDD	VDD	0	1	ZINV_133_inst_52482
1216.57	222.6	INV_X4	VSS	VSS	0	1	ZINV_133_inst_52482
1215.62	85.4	INV_X2	VDD	VDD	0	1	ZINV_49_inst_52486
1215.62	85.4	INV_X2	VSS	VSS	0	1	ZINV_49_inst_52486
1215.62	110.6	INV_X2	VDD	VDD	0	1	ZINV_51_inst_52489
1215.62	110.6	INV_X2	VSS	VSS	0	1	ZINV_51_inst_52489
1216.19	110.6	INV_X2	VDD	VDD	0	1	ZINV_51_inst_52492
1216.19	110.6	INV_X2	VSS	VSS	0	1	ZINV_51_inst_52492
1216.76	110.6	INV_X2	VDD	VDD	0	1	ZINV_51_inst_52495
1216.76	110.6	INV_X2	VSS	VSS	0	1	ZINV_51_inst_52495
.....	.....	.....	...	....	...	....	.....
.....	.....	.....	...	....	...	....	.....
.....	.....	.....	...	....	...	....	.....

©2020 ANSYS, Inc. / Confidential

Disconnected Instance Reports: Argument(s)

`data_integrity_reports.write_unconnected_pg_pins_report`

Required Argument
ev

Optional Arguments	
output_file	“./unconnected_instance_pin.rpt”
columns	
formats	
header	
footer	
max_lines	5000

©2020 ANSYS, Inc. / Confidential

Disconnected Instance Reports: `columns`

```
data_integrity_reports.write_unconnected_pg_pins_report
```

Valid options for `columns`
`loc_x`
`loc_y`
`cell_name`
`pin`
`net`
`logical_disconnect`
`physical_disconnect`
`instance`

©2020 ANSYS, Inc. / Confidential

Disconnected Instance Report: Usage Examples

Report all the disconnected instance pins in the design

```
data_integrity_reports.write_unconnected_pg_pins_report(ev, max_lines=None)
```


©2020 ANSYS, Inc. / Confidential

Construction Check Reports: Shorted Nodes

`data_integrity_reports.write_short_report`

```
data_integrity_reports.write_short_reports(ev)
```

#Net_1	Net_2	Layer	Location
VSS	core3/VDD_INT	metal1	653.2150, 1.4525
VSS	core3/VDD_INT	metal1	660.8850, 1.4450
VSS	core3/VDD_INT	metal1	660.8850, 1.4775
VSS	core3/VDD_INT	metal1	690.8850, 1.4525
VSS	core3/VDD_INT	metal1	720.8850, 1.4450
VSS	core3/VDD_INT	metal1	720.8850, 1.4775
VSS	core3/VDD_INT	metal1	724.4925, 1.4525
VSS	core3/VDD_INT	metal1	726.1000, 1.4450
VSS	core3/VDD_INT	metal1	841.2500, 1.4450
VSS	core3/VDD_INT	metal1	726.1000, 1.4775
VSS	core3/VDD_INT	metal1	732.4925, 1.4775
VSS	core3/VDD_INT	metal1	740.8850, 1.4775
VSS	core3/VDD_INT	metal1	770.8850, 1.4775
...	.....	.....	.....
...	.....	.....	.....
...	.....	.....	.....

©2020 ANSYS, Inc. / Confidential

Shorted Node Report: Argument(s)

`data_integrity_reports.write_short_report`

Required Argument
ev

Optional Arguments	
output_file	“./unconnected_instance_pin.rpt”

©2020 ANSYS, Inc. / Confidential

Construction Check Reports: Disconnected Nodes

`data_integrity_reports.write_unconnected_node_report`

```
data_integrity_reports.write_unconnected_node_report(ev)
```

```
#Total disconnected nodes = 5780 . Line limit used = 100000
#    X          Y        Layer                Net
  944.3675   796.1800     metal1              VDD
 1218.6600     2.8000     metal1              VDD
 1223.2200     2.8000     metal1              VDD
 1225.5000     2.8000     metal1              VDD
 1226.3550     2.8000     metal1              VDD
 1226.5450     2.8000     metal1              VDD
 1226.7350     2.8000     metal1              VDD
 1218.6600     5.6000     metal1              VDD
 1223.2200     5.6000     metal1              VDD
 1225.5000     5.6000     metal1              VDD
 1226.3550     5.6000     metal1              VDD
 1226.5450     5.6000     metal1              VDD
 1226.7350     5.6000     metal1              VDD
 ..........................................
 ..........................................
 ..........................................
```


©2020 ANSYS, Inc. / Confidential

Disconnected Node Report: Argument(s)

`data_integrity_reports.write_unconnected_node_report`

Required Argument
ev

Optional Arguments	
output_file	“./disconnected_nodes.rpt”
limit	100,000

©2020 ANSYS, Inc. / Confidential

Power/Voltage/Current/Resistance Reports

emir_reports

- Power Reports

- Voltage/Current Reports
  - Bump Current/Voltage Reports
  - Demand/Supply Current Reports
  - Instance Voltage Reports
  - Layer Voltage Report/Node Voltage Report
  - Switch Voltage Report
  - Electromigration Report

- Resistance Reports
  - SPR Report
  - Fast Effective Resistance Report

©2020 ANSYS, Inc. / Confidential

Power Reports

`emir_reports.write_instance_power_report_and_summary`

```
**** RedHawk-SC Power Summary Report ****
Created: Mon May  4 23:52:30 2020

A total of 1185384 instances were summarized for this report while 807905 were omitted due to missing power data (31.84% 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                 0.061068            0.038358           0.010882          0.068866            0.11808         82.16             282689     282689
core3/VDD_INT       0.0085734           0.0046497          0.003602          0.018233            0.025648        17.84             94790      94790
Total               0.069641            0.043008           0.014484          0.087098            0.14373         100.00            377479     377479

*** Frequency Domains:
1.25e+08            0.069641            0.042917           0.014484          0.087098            0.14373         100.00            377479     377479
Total               0.069641            0.042917           0.014484          0.087098            0.14373         100.00            377479     377479

*** User Defined Groups:
combinational logic 0                   0.0060018          0.0087147         0.083258            0.096534        67.17             305671     305671
sequential logic    0.069641            0.037575           0.0057252         0.0038403           0.047141        32.80             71800      71800
memory              0                   7.6725e-06         4.43e-05          0                   5.1973e-05      0.04              8          8
Total               0.069641            0.043585           0.014484          0.087098            0.14373         100.00            377479     377479

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

```
>>> summary = emir_reports.write_instance_power_report_and_summary(scn)
>>> summary
{'clock': {'Total': {'percent_power': 0}},
'domain': {"Net('VDD')": {'clock_pin_power': 0.06106774578938712,
                          'instance_count': 282689,
                          'internal_power': 0.0383582320820679,
                          'leakage_power': 0.010882196835498625,
                          'percent_power': 82.15531222478005,
                          'pin_count': 282689,
                          'switching_power': 0.0688658116659866,
                          'total_power': 0.11808100451804293},
           "Net('core3/VDD_INT')": {'clock_pin_power': 0.008573434120080492,
                                    'instance_count': 94790,
                                    'internal_power': 0.004649668683926134,
                                    'leakage_power': 0.003602008892117148,
                                    'percent_power': 17.84468777521995,
                                    'pin_count': 94790,
                                    'switching_power': 0.01823250459287351,
                                    'total_power': 0.025647990382455 8},
```


©2020 ANSYS, Inc. / Confidential

Power Reports: Argument(s)

`emir_reports.write_instance_power_report_and_summary`

Required Argument
view

Optional Arguments	
file_name_detailed_report	“./power.rpt”
file_name_summary_report	“./power_summary.rpt”
columns	
sort	True
sort_order	‘descending’
sort_columns	[‘total_power’]
header_instance_report	
header_summary_report	
format_instance_report	
format_summary_report	
max_lines_detailed_report	5000
write_instance_file	False

©2020 ANSYS, Inc. / Confidential

Power Reports: columns

```
emir_reports.write_instance_power_report_and_summary
```

Allowed values for columns
instance
cell_name
loc_x
loc_y
pin
domain
clock_pin_power
frequency

Allowed values for columns
internal_power
leakage_power
source
switching_power
toggle_rate
total_power
voltage

©2020 ANSYS, Inc. / Confidential

Power Reports: Creating Instance Power File

```
emir_reports.write_instance_power_report_and_summary(scn,
write_instance_file=True, max_lines_detailed_report=None)
```

```
# 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          4.075e-06       4.321e-07       7.132e-05    7.582e-05    1.10    INV_X32            cts_inv_525660807
  VDD     VDD     1.25e+08     2.00             0          1.028e-06        2.16e-07       8.136e-05    8.261e-05    1.10    INV_X16            cts_inv_526160812
  VDD     VDD     1.25e+08     2.00             0          4.049e-06       4.321e-07        7.4e-05     7.848e-05    1.10    INV_X32            cts_inv_526560816
  VDD     VDD     1.25e+08     2.00             0          9.244e-07        2.16e-07       6.886e-05      7e-05      1.10    INV_X16            cts_inv_527260823
  VDD     VDD     1.25e+08     2.00             0          1.016e-06       2.433e-07       6.484e-05     6.61e-05    1.10    INV_X16            cts_inv_584061391
  VDD     VDD     1.25e+08     2.00             0         -1.962e-07        2.16e-07       6.138e-05     6.14e-05    1.10    INV_X16            cts_inv_527360824
  VDD     VDD     1.25e+08     2.00             0          1.399e-06       2.433e-07       5.751e-05    5.915e-05    1.10    INV_X16            cts_inv_584561396
  VDD     VDD     1.25e+08     2.00             0          9.803e-07       2.433e-07       6.954e-05    7.076e-05    1.10    INV_X16            cts_inv_584661397
  VDD     VDD     1.25e+08     2.00             0          1.502e-06       2.433e-07       4.663e-05    4.837e-05    1.10    INV_X16            cts_inv_584761398
  VDD     VDD     1.25e+08     2.00             0          2.006e-06       2.433e-07       5.574e-05    5.799e-05    1.10    INV_X16            cts_inv_585661407
  VDD     VDD     1.25e+08     2.00             0          1.676e-06       2.433e-07       5.136e-05    5.328e-05    1.10    INV_X16            cts_inv_585961410
  VDD     VDD     1.25e+08     2.00             0          1.012e-06       2.433e-07       5.961e-05    6.087e-05    1.10    INV_X16            cts_inv_586161412
  VDD     VDD     1.25e+08     2.00             0          2.006e-06       2.433e-07       5.653e-05    5.878e-05    1.10    INV_X16            cts_inv_586561416
  VDD     VDD     1.25e+08     2.00             0         -4.219e-07       1.216e-07       4.462e-05    4.432e-05    1.10    INV_X8             inv_drc_cln62326
  VDD     VDD     1.25e+08     2.00             0         -7.325e-07        1.08e-07       3.476e-05    3.414e-05    1.10    INV_X8             cts_inv_525860809
  VDD     VDD     1.25e+08     2.00             0         -2.786e-07       2.433e-07       6.638e-05    6.634e-05    1.10    INV_X16            cts_inv_526360814
  VDD     VDD     1.25e+08     1.00             0         -2.895e-06        8.52e-08        3.28e-05    2.999e-05    1.10    BUF_X4             ZBUF_62_inst_51475
  VDD     VDD     1.25e+08     2.00             0          7.505e-07       2.698e-08       3.606e-06    4.384e-06    1.10    CLKBUF_X3          ZBUF_314_inst_65462
```


©2020 ANSYS, Inc. / Confidential

Power Reports: Usage Examples

Customize the columns reported in the Instance Power File

```
columns = ['pin', 'domain', 'total_power', 'instance']
formats = '{0:10} {1:10} {2:10.4f} {3}'
header  = '{0:10} {1:10} {2:10} {3}'.format(*columns)

emir_reports.write_instance_power_report_and_summary(
   scn, write_instance_file=True, columns=columns,
   format_instance_report=formats, header_instance_report=header)
```


©2020 ANSYS, Inc. / Confidential

Bump Voltage/Current Reports

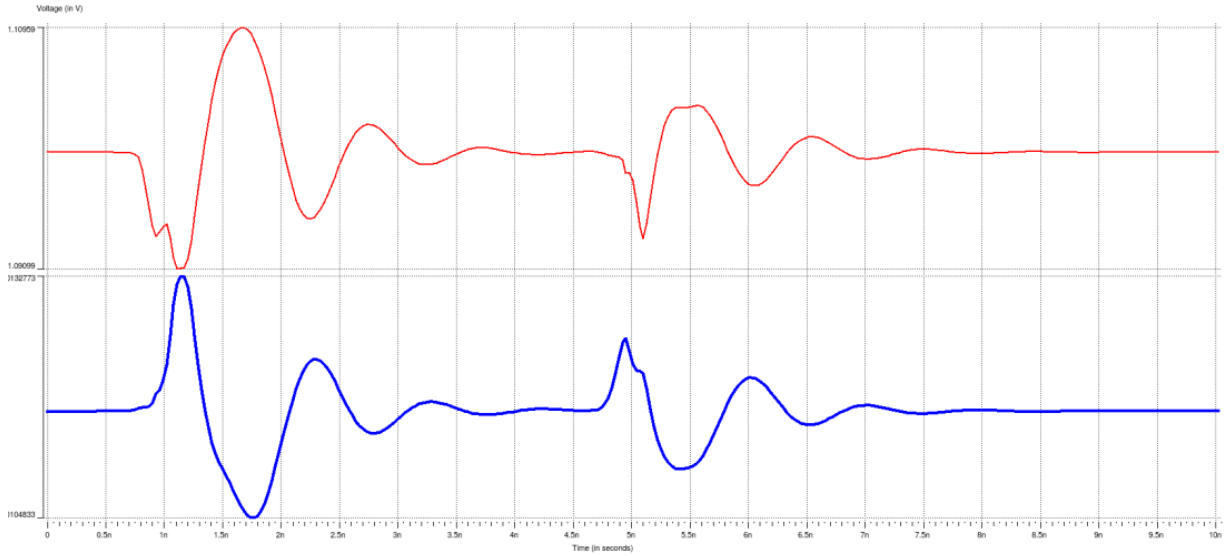
```
emir_reports.write_bump_currents
emir_reports.write_bump_voltages
```

```
TitleText: Bump Currents
# Time (ps) I (A)
"VSS_1
        0.00    -0.0000001
       30.00    -0.0000002
       60.00    -0.0000001
       90.00     0.0000000
      120.00     0.0000003
      150.00     0.0000004
       ......    .........
       ......    .........

"VSS_10
        0.00    -0.0000356
       30.00    -0.0000356
       60.00    -0.0000355
       90.00    -0.0000354
      120.00    -0.0000352
      150.00    -0.0000351
       ......    .........
       ......    .........

"VDD_1
        0.00     0.0000319
       30.00     0.0000319
       60.00     0.0000319
       90.00     0.0000318
      120.00     0.0000318
      150.00     0.0000318
       ......    .........
       ......    .........

"VDD_10
        0.00     0.0000448
       30.00     0.0000448
       60.00     0.0000448
       90.00     0.0000448
      120.00     0.0000448
      150.00     0.0000448
       ......    .........
       ......    .........
```



VDD/VSS Bump Voltage in SignalViewer

```
TitleText: Bump Voltages
# Time (ps) V (V)
"VSS_1
        0.00     0.0000000
       30.00     0.0000001
       60.00     0.0000000
       90.00    -0.0000002
      120.00    -0.0000007
      150.00    -0.0000013
       ......    .........
       ......    .........

"VSS_10
        0.00     0.0000000
       30.00     0.0000000
       60.00    -0.0000000
       90.00    -0.0000003
      120.00    -0.0000007
      150.00    -0.0000012
       ......    .........
       ......    .........

"VDD_1
        0.00     1.1000000
       30.00     1.1000000
       60.00     1.1000000
       90.00     1.1000000
      120.00     1.1000000
      150.00     1.1000000
       ......    .........
       ......    .........

"VDD_10
        0.00     1.1000000
       30.00     1.1000000
       60.00     1.1000000
       90.00     1.1000000
      120.00     1.1000000
      150.00     1.1000000
       ......    .........
       ......    .........
```


©2020 ANSYS, Inc. / Confidential

Bump Voltage/Current Reports

```
emir_reports.write_bump_currents
emir_reports.write_bump_voltages
```

Required Argument
analysis_view

Optional Argument	
output_file	“./bump_currents.sv”

Optional Argument	
output_file	“./bump_voltages.sv”

©2020 ANSYS, Inc. / Confidential

Demand/Supply Current Reports

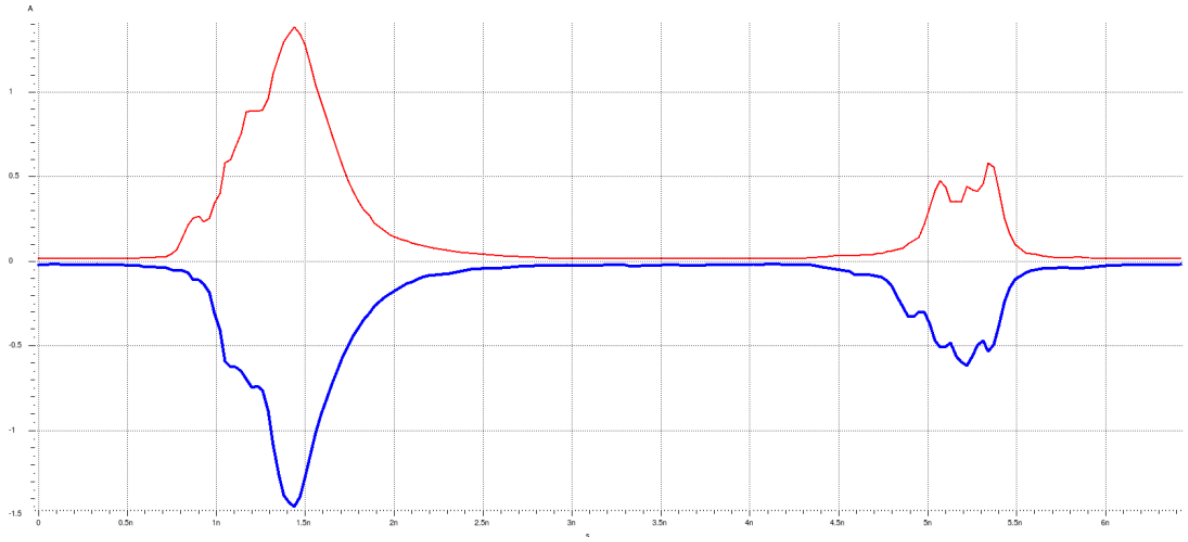
```
emir_reports.write_demand_currents
emir_reports.write_supply_currents
```

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

"core3/VDD_INT
        0.00      0.0038027
       60.00      0.0044067
       90.00      0.0040929
      120.00      0.0047936
      150.00      0.0041823
      ......      .........
      ......      .........

"VSS
        0.00     -0.0212973
       60.00     -0.0185013
      120.00     -0.0180928
      150.00     -0.0202592
      ......      .........
      ......      .........

"VDD
        0.00      0.0136936
      480.00      0.0149988
      540.00      0.0161063
      570.00      0.0208575
      600.00      0.0219171
      ......      .........
      ......      .........
```



VDD/VSS Demand Currents in SignalViewer

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

"VSS
        0.00     -0.0212973
       30.00     -0.0212430
       60.00     -0.0210722
       90.00     -0.0208145
      120.00     -0.0204673
      150.00     -0.0201480
      ......      .........
      ......      .........

"VDD
        0.00      0.0136936
       30.00      0.0137001
       60.00      0.0137303
       90.00      0.0137670
      120.00      0.0138323
      150.00      0.0139005
      ......      .........
      ......      .........
```


©2020 ANSYS, Inc. / Confidential

Bump Voltage/Current Reports

```
emir_reports.write_demand_currents
emir_reports.write_supply_currents
```

Required Argument
analysis_view

Optional Argument	
output_file	“./demand_current.sv”

Optional Argument	
output_file	“./supply_current.sv”

©2020 ANSYS, Inc. / Confidential

Instance Voltage Report

`emir_reports.write_all_instance_voltages`

```
>>> emir_reports.write_all_instance_voltages(av_dynamic)
```

loc_x (u)	loc_y (u)	eff_Vdd (v)	max_pg_tw (v)	min_pg_tw (v)	min_pg_sim (v)	max_pg_sim (v)	min_vdd_tw (v)	max_vss_tw (v)	pg_arc pwr/gnd	instance
894.735	518.105	0	1.104	1.081	1.016	1.114	1.09	0.00878	core3/VDD_INT/VSS	core3/regfile_program_memory.DFF_X1_5801
894.355	518.105	1.078	1.101	1.018	1.018	1.114	1.101	0.08243	core3/VDD_INT/VSS	core3/cts_inv_529824785
898.725	518.105	0	1.104	1.081	1.018	1.114	1.09	0.008812	core3/VDD_INT/VSS	core3/regfile_program_memory.DFF_X1_620
39.9	1141.0	1.069	1.1	1.02	1.02	1.123	1.025	0.005166	VDD/VSS	cts_inv_551661067
41.61	1143.8	1.073	1.1	1.072	1.02	1.124	1.087	0.01495	VDD/VSS	core1.regfile_data_memory.DFF_X1_5956
893.785	518.105	1.084	1.114	1.081	1.021	1.114	1.09	0.008772	core3/VDD_INT/VSS	core3/regfile_program_memory.NAND3_X1_302
37.81	1143.8	1.074	1.1	1.074	1.022	1.124	1.089	0.01513	VDD/VSS	core1.regfile_data_memory.DFF_X1_1621
894.165	518.105	0	1.114	1.082	1.022	1.114	1.091	0.00865	core3/VDD_INT/VSS	core3/regfile_program_memory.NAND2_X1_940
46.74	1143.8	0	1.124	1.072	1.023	1.124	1.087	0.01477	VDD/VSS	core1.regfile_data_memory.MUX2_X1_4796
279.3	732.2	0	1.1	1.086	1.024	1.121	1.094	0.007483	VDD/VSS	core1.regfile_program_memory.INV_X1_3086
275.69	729.4	1.074	1.1	1.024	1.024	1.121	1.029	0.005383	VDD/VSS	cts_inv_558261133
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....

©2020 ANSYS, Inc. / Confidential

Instance Voltage Report: Arguments

`emir_reports.write_all_instance_voltages`

Required Argument
av

Optional Arguments	
output_file	“./voltage_stats.rpt”
columns	
sort	True
sort_order	descending
sort_columns	[‘min_pg_sim’]
formats	
header	
footer	
max_lines	5000
domain_filter	None
skip_instances_with_no_effdvd	False
report_macros	False

©2020 ANSYS, Inc. / Confidential

Instance Voltage Report: columns

```
emir_reports.write_all_instance_voltages
```

Valid column names
loc_x
loc_y
eff_Vdd
max_pg_tw
min_pg_tw
min_pg_sim**
max_pg_sim

Valid column names
min_vdd_tw
max_vss_tw
pg_arc
pin_pg_arc*
cell_name*
instance

* Not printed out by default
** Available only for a Static AnalysisView

©2020 ANSYS, Inc. / Confidential

Instance Voltage Report: Report all instances in the design

```
emir_reports.write_all_instance_voltages(av, max_lines=None, sort=False)
```

# loc_x (u)	loc_y (u)	eff_Vdd (v)	max_pg_tw (v)	min_pg_tw (v)	min_pg_sim (v)	max_pg_sim (v)	min_vdd_tw (v)	max_vss_tw (v)	pg_arc pwr/gnd	instance
894.735	518.105	0	1.104	1.081	1.016	1.114	1.09	0.00878	core3/VDD_INT/VSS	core3/regfile_program_memory.DFF_X1_5801
894.355	518.105	1.078	1.101	1.018	1.018	1.114	1.101	0.08243	core3/VDD_INT/VSS	core3/cts_inv_529824785
898.725	518.105	0	1.104	1.081	1.018	1.114	1.09	0.008812	core3/VDD_INT/VSS	core3/regfile_program_memory.DFF_X1_620
39.9	1141.0	1.069	1.1	1.02	1.02	1.123	1.025	0.005166	VDD/VSS	cts_inv_551661067
41.61	1143.8	1.073	1.1	1.072	1.02	1.124	1.087	0.01495	VDD/VSS	core1.regfile_data_memory.DFF_X1_5956
893.785	518.105	1.084	1.114	1.081	1.021	1.114	1.09	0.008772	core3/VDD_INT/VSS	core3/regfile_program_memory.NAND3_X1_302
37.81	1143.8	1.074	1.1	1.074	1.022	1.124	1.089	0.01513	VDD/VSS	core1.regfile_data_memory.DFF_X1_1621
894.165	518.105	0	1.114	1.082	1.022	1.114	1.091	0.00865	core3/VDD_INT/VSS	core3/regfile_program_memory.NAND2_X1_940
46.74	1143.8	0	1.124	1.072	1.023	1.124	1.087	0.01477	VDD/VSS	core1.regfile_data_memory.MUX2_X1_4796
279.3	732.2	0	1.1	1.086	1.024	1.121	1.094	0.007483	VDD/VSS	core1.regfile_program_memory.INV_X1_3086
275.69	729.4	1.074	1.1	1.024	1.024	1.121	1.029	0.005383	VDD/VSS	cts_inv_558261133
.....	.....	.....	...	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	...	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	...	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	...	.....	.....	.....	.....	.....	.....	.....

©2020 ANSYS, Inc. / Confidential

Instance Voltage Report: Usage Examples

Customize the columns reported in the Voltage Report File

```
columns = ['loc_x', 'loc_y', 'min_pg_sim', 'pin_pg_arc', 'instance']
formats = '{0:10} {1:10} {2:20.5f}    {3:20} {4}'
header  = '{0:10} {1:10} {2:20}    {3:20} {4}\n'.format(*columns)

emir_reports.write_all_instance_voltages(av, columns=columns, formats=formats,
                                         header=header, max_lines=1000)
```


©2020 ANSYS, Inc. / Confidential

Instance Voltage Report: Usage Examples

Sorting the output report by a different column

```
emir_reports.write_all_instance_voltages(av, max_lines=1000,
                                         report_macros=True, sort=True,
                                         sort_columns=['min_pg_tw', 'min_pg_sim'])
```


©2020 ANSYS, Inc. / Confidential

Instance Voltage Report: Usage Examples

Using `domain_filter` to set voltage thresholds for reporting

```
domain_filter = { Net('core3/VDD_INT'): {'dynamic':0.07, 'static':0.01},
                  Net('VDD')          : {'dynamic':0.05,},
                  'default'           : {'static':0.03, 'dynamic':0.1} }

emir_reports.write_all_instance_voltages(av, max_lines=1000, sort=True
                                         skip_instances_with_no_effdvd=True,
                                         domain_filter=domain_filter,
                                         report_macros=True)
```


©2020 ANSYS, Inc. / Confidential

Instance Voltage Report: Full Custom Reports

- `write_all_instance_voltages` is a convenience API over a fast and efficient data structure

- The `ChunkedData` Data Structure is directly accessible to the user
  - Easy to create fast, custom reports
  - Can be saved and reloaded from the `SeaScapeDB`
  - Missing fields in the report can be added
  - Helps in analytics across multiple AnalysisViews

©2020 ANSYS, Inc. / Confidential

Interlude: Creating and Accessing a `ChunkedData` Object

```
def create_voltage_info(av):
    dv = av.get_related_views(DesignView)[0]
    mm = gp.MapReduce(dv)
    voltage_info = ChunkedData()
    mm.map_reduce(dv.get_mr_instances(), partial(_get_voltage_info, av=av,
                                                 voltage_info=voltage_info),
                  finalize_data=lambda xx:xx.finalize())
    return mm

voltage_data_mr = create_voltage_info(av)
voltage_data = voltage_data_mr.get()
```

```
def _get_voltage_info(instances, av, voltage_info):
    dv = av.get_related_views(DesignView)[0]
    items = dict()
    for instance in instances:
        try:
            effdvd = av.get_voltage_stats(instance).get_fullsim_effdvd()
        except ProbeError:
            continue
        try:
            oversim = av.get_voltage_stats(instance).get_fullsim()
        except ProbeError:
            continue
        items[instance] = (effdvd.get_min(), oversim.get_min())
    partition_id = dv.get_partition_id(instances)
    voltage_info.add_chunk_data(partition_id, items)
    return voltage_info
```


©2020 ANSYS, Inc. / Confidential

Instance Voltage Report: Creating the `ChunkedData` Container

```
data = emir_reports.get_instance_voltage_data(av)
voltage_data = data.get()
```


©2020 ANSYS, Inc. / Confidential

Instance Voltage Report: Instance Voltage Data

The parts of the ChunkedData are organized as a dict of dicts of dicts in the following format

```
part_data[<Instance>][<pgarc tuple>]['cell_id'] = Integer representing the ID of the Cell
part_data[<Instance>][<pgarc tuple>]['ideal_voltage'] = Ideal voltage of Net attached to the Power pin
part_data[<Instance>][<pgarc tuple>]['power_net'] = Net connected to Power Pin
part_data[<Instance>][<pgarc tuple>]['ground_net'] = Net connected to Net Pin
part_data[<Instance>][<pgarc tuple>]['coord'] = <Instance>.get_coord()

    effdvd = <voltage arc stat>.get_fullsim_effdvd()
part_data[<Instance>][<pgarc tuple>]['effdvd'] = effdvd.get_min()
part_data[<Instance>][<pgarc tuple>]['effdvd_vdd_avg'] = effdvd.get_vdd_at_min()
part_data[<Instance>][<pgarc tuple>]['effdvd_vss_avg'] = effdvd.get_vss_at_min()

    mintw = <voltage arc stat>.get_fullsim_over_tw()
part_data[<Instance>][<pgarc tuple>]['mintw'] = mintw.get_min()
part_data[<Instance>][<pgarc tuple>]['mintw_vss_min'] = mintw.get_vss_at_min()
part_data[<Instance>][<pgarc tuple>]['mintw_vdd_min'] = mintw.get_vdd_at_min()

    oversim = <voltage arc stat>.get_fullsim()
part_data[<Instance>][<pgarc tuple>]['oversim'] = oversim.get_min()
part_data[<Instance>][<pgarc tuple>]['oversim_vss_min'] = oversim.get_vss_at_min()
```


©2020 ANSYS, Inc. / Confidential

Instance Voltage Report: Creating a Custom Report

```
def _seq_report(dv, ivolt_chunked, chunk_part_id, fp):
    fp.initialize_part(str(chunk_part_id))
    data_dict = ivolt_chunked.get_chunk_data(chunk_part_id)
    results = list()
    for instance, pgarc_dict in part_data.iteritems():
        cell_id = instance.get_cell_id()
        cell_type = dv.get_cell_type(Cell(cell_id))
        if not cell_type.is_seq_cell():
            continue
        for pgarc, data_dict in pgarc_dict.iteritems():
            eff_dvd = data_dict['effdvd']
            min_tw = data_dict['mintw']
            min_wc = data_dict['oversim']
            results.append((instance, eff_dvd, min_tw, min_wc))
    for result in dv.convert_to_name(results):
        fp.write('{0:50} {1:7.4f} {2:7.4f} {3:7.4f}\n'.format(*result))
    fp.close()
    return fp
```

```
def process_voltage_data(dv, file_name, chunked_data):
    fp = gp.gp_distributed_file(file_name)
    mm = gp.MapReduce(dv)
    for chunk_id in chunked_data.get_chunks():
        mm.map_part(partial(_seq_report, dv=dv, ivolt_chunked=chunked_data,
                            chunk_part_id=chunk_id, fp=fp))
    mm.reduce()


ivolt_d = gp_delayed_object(emir_reports.get_instance_voltage_data(av))
ivolt_chunked = ivolt_d.get()
process_voltage_data(dv, './my_seq_cells.rpt', ivolt_chunked)
```

Example showing a method to generate the voltage report for only sequential cells

©2020 ANSYS, Inc. / Confidential

Layer Voltage Report

emir_reports.write_layer_voltage_report

```
summary = emir_reports.write_layer_voltage_report(av_dynamic)
```

# min_x (u)	min_y (u)	min_voltage_drop (v)	max_x (u)	max_y (u)	max_voltage_drop (v)	layer_drop (v)	net	layer
1047.54	617.4	0.004531	896.72	518.105	0.08243	0.0779	VSS	metal1
1157.005	579.6	0.004114	43.025	1142.4	0.07452	0.07041	VDD	metal1
1178.0	0.0	0.00163	71.6	64.8	0.01449	0.01286	VSS	metal12
1175.14	578.2	0.005415	31.615	7.0	0.01762	0.0122	VSS	metal2
1174.94	578.2	0.005437	31.74	7.0	0.01748	0.01204	VSS	metal3
1174.94	578.2	0.005451	31.74	7.0	0.01737	0.01192	VSS	metal4
1175.14	578.2	0.005459	31.69	7.0	0.01732	0.01186	VSS	metal5
1175.14	578.2	0.005467	31.69	7.0	0.01726	0.01179	VSS	metal6
1172.5	578.2	0.005473	33.34	7.0	0.01721	0.01174	VSS	metal7
1175.02	578.2	0.005497	31.66	7.0	0.01699	0.01149	VSS	metal8
1172.08	578.2	0.005528	32.08	7.0	0.01672	0.01119	VSS	metal9
1172.5	572.6	0.005542	33.34	1.4	0.01662	0.01108	VSS	metal10
975.14	599.2	0.005381	415.14	918.4	0.01644	0.01106	VDD	metal2
1178.8	613.6	0.005506	33.34	61.6	0.0158	0.0103	VSS	metal11
1040.0	0.0	0.002128	28.0	1080.53	0.01235	0.01022	VDD	metal12
974.94	599.2	0.005514	414.94	918.4	0.0154	0.009882	VDD	metal3
974.94	599.2	0.005606	414.94	918.4	0.01461	0.009008	VDD	metal4
975.14	599.2	0.005656	415.14	918.4	0.01415	0.008498	VDD	metal5
975.14	599.2	0.005706	415.14	918.4	0.0137	0.007995	VDD	metal6
974.69	599.2	0.005748	414.69	918.4	0.01329	0.007538	VDD	metal7
1039.205	539.105	0.005851	13.76	1122.8	0.01196	0.00611	VDD	metal8
1042.4	567.6	0.006017	27.2	1124.4	0.01213	0.00611	VDD	metal11
1039.625	525.105	0.005907	14.18	1122.8	0.01197	0.006063	VDD	metal9
1041.725	525.105	0.00593	14.18	1122.8	0.01197	0.006044	VDD	metal10

©2020 ANSYS, Inc. / Confidential

Layer Voltage Reports: Argument(s)

`emir_reports.write_layer_voltage_report`

Required Argument
analysis_view

Optional Arguments	
output_file	“./layer_voltage.rpt”
columns	
sort	True
sort_order	‘descending’
sort_columns	[‘layer_drop’]
formats	
header	
footer	
report_type	‘relative’

©2020 ANSYS, Inc. / Confidential

Layer Voltage Report: columns

emir_reports.write_layer_voltage_report

Valid column names
min_x
min_y
max_x
max_y

Valid column names
net
layer
min_voltage
max_voltage
layer_drop

©2020 ANSYS, Inc. / Confidential

Layer Voltage Report: Usage Examples

Customize the columns reported in the Layer Voltage Report File

```
columns = ['min_x', 'min_y', 'min_voltage', 'layer_drop', 'net', 'layer']
formats = '{0:10} {1:10} {2:10.6f} {3:20.6f} {4:20} {5}'
header  = '{0:10} {1:10} {2:10} {3:20} {4:20} {5}\n'.format(*columns)
sort_by = ['net', 'min_voltage']

emir_reports.write_layer_voltage_report(av, columns=columns, formats=formats,
                                        header=header, sort_columns=sort_by)
```


©2020 ANSYS, Inc. / Confidential

Node Voltage Report

`emir_reports.write_node_voltage_report`

```
emir_reports.write_node_voltage_report(av_dynamic)
```

```
# contents are sorted in decending order by layer, net, node_voltage
# loc_x      loc_y      layer       net     node_voltage
#  (u)        (u)                                (v)
   71.6       64.8     metal12      VSS        0.01449
   76.4       64.8     metal12      VSS        0.01448
   71.6       63.2     metal12      VSS        0.01448
   73.2       64.8     metal12      VSS        0.01447
   76.4       63.2     metal12      VSS        0.01447
   74.8       64.8     metal12      VSS        0.01447
   73.2       63.2     metal12      VSS        0.01446
   74.8       63.2     metal12      VSS        0.01446
   74.0       64.8     metal12      VSS        0.01446
   73.2       66.4     metal12      VSS        0.01445
   74.8       66.4     metal12      VSS        0.01445
   76.4       66.4     metal12      VSS        0.01445
   71.6       66.4     metal12      VSS        0.01445
   74.0       66.4     metal12      VSS        0.01445
   74.0       63.2     metal12      VSS        0.01444
   73.2       61.6     metal12      VSS        0.01441
   74.8       61.6     metal12      VSS        0.01441
   74.0       61.6     metal12      VSS        0.01441
   76.4       61.6     metal12      VSS        0.01441
   ....       ....     .......      ...        .......
   ....       ....     .......      ...        .......
```


©2020 ANSYS, Inc. / Confidential

Node Voltage Reports: Argument(s)

`emir_reports.write_node_voltage_report`

Required Argument
av

Optional Arguments	
output_file	"./node_voltage.rpt"
columns	
sort	True
sort_order	'descending'
sort_columns	['layer', 'net', 'node_voltage']
formats	
header	
footer	
max_lines	5000
report_type	'relative'

©2020 ANSYS, Inc. / Confidential

Node Voltage Report: `columns`

`emir_reports.write_node_voltage_report`

Valid column names
`loc_x`
`loc_y`
`net`
`layer`
`node_voltage`

©2020 ANSYS, Inc. / Confidential

Node Voltage Report: Usage Examples

Customize the columns reported in the Node Voltage Report File

```
columns = ['loc_x', 'loc_y', 'net', 'layer', 'node_voltage']
formats = '{0:10} {1:10} {2:20} {3:20} {4:10.5f}'
header  = '{0:10} {1:10} {2:20} {3:20} {4:10}\n'.format(*columns)
sort_by = ['node_voltage']

emir_reports.write_node_voltage_report(av, columns=columns, formats=formats,
                                       header=header, sort_columns=sort_by)
```


©2020 ANSYS, Inc. / Confidential

Switch Voltage Report

`emir_reports.write_switch_report`

```
emir_reports.write_switch_report(av_dynamic)
```

#instance_name	internal_pin_Voltage	switch_voltage	idsat
core3/inst_sw_S1_0	1.1075	0.0002	1.712463e-04
core3/inst_sw_S1_7	1.1076	0.0002	2.173246e-04
core3/inst_sw_S1_14	1.1076	0.0002	2.020508e-04
core3/inst_sw_S1_21	1.1076	0.0002	1.732654e-04
core3/inst_sw_S1_28	1.1075	0.0002	1.577121e-04
core3/inst_sw_S1_35	1.1075	0.0002	2.106169e-04
core3/inst_sw_S1_42	1.1075	0.0003	2.245411e-04
core3/inst_sw_S1_49	1.1075	0.0003	2.038293e-04
core3/inst_sw_S1_56	1.1074	0.0003	2.163459e-04
core3/inst_sw_S1_63	1.1074	0.0003	1.750101e-04
core3/inst_sw_S1_70	1.1075	0.0002	1.554035e-04
core3/inst_sw_S1_77	1.1075	0.0002	2.719516e-04
core3/inst_sw_S1_84	1.1073	0.0004	2.900841e-04
core3/inst_sw_S1_91	1.1073	0.0004	1.779328e-04
core3/inst_sw_S1_98	1.1072	0.0005	1.569088e-04
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....

©2020 ANSYS, Inc. / Confidential

Switch Voltage Report: Argument(s)

```
emir_reports.write_switch_report
```

Required Argument
av

Optional Argument	
output_file	“./switch_voltage_stats.rpt

©2020 ANSYS, Inc. / Confidential

Electromigration Reports: Metal Segments

`emir_reports.write_em_metal_report`

```
emir_reports.write_em_metal_report(emv)
```

```
# em_type = DC, ignore_sliver = True, 'length' column is blech length
#  layer            from                  to              length     width      current   constraint violation  status net
#                    (u)                   (u)              (u)       (u)        (A)        (A)       (%)
  metal1      (662.46,197.505)      (662.885,197.505)      84.19      0.07    0.0002431  0.0006909    35.19     PASS  core3/VDD_INT
  metal1      (662.27,197.505)       (662.46,197.505)      84.19      0.07    0.0002431  0.0006909    35.19     PASS  core3/VDD_INT
  metal1     (661.795,197.505)      (662.27,197.505)       84.19      0.07    0.0002431  0.0006909    35.19     PASS  core3/VDD_INT
  metal1     (660.885,197.505)     (661.795,197.505)       84.19      0.07    0.0002431  0.0006909    35.19     PASS  core3/VDD_INT
  metal1     (662.885,197.505)      (663.25,197.505)       84.19      0.17    0.0002431   0.001738    13.99     PASS  core3/VDD_INT
  metal1      (663.25,197.505)      (663.595,197.505)      84.19      0.17    0.0002428   0.001738    13.97     PASS  core3/VDD_INT
  metal1     (663.595,197.505)     (664.0175,197.505)      84.19      0.17    0.0002428   0.001738    13.97     PASS  core3/VDD_INT
  metal1     (664.0175,197.505)     (664.2,197.505)        84.19      0.17    0.0002425   0.001738    13.96     PASS  core3/VDD_INT
  metal1      (664.2,197.505)      (664.545,197.505)       84.19      0.17    0.0002425   0.001738    13.95     PASS  core3/VDD_INT
  metal1      (666.07,197.505)     (666.425,197.505)       84.19      0.17    0.0002422   0.001738    13.94     PASS  core3/VDD_INT
  metal1      (665.86,197.505)      (666.07,197.505)       84.19      0.17    0.0002422   0.001738    13.94     PASS  core3/VDD_INT
  metal1      (665.68,197.505)      (665.86,197.505)       84.19      0.17    0.0002422   0.001738    13.94     PASS  core3/VDD_INT
  metal1     (665.305,197.505)     (665.495,197.505)       84.19      0.17    0.0002422   0.001738    13.94     PASS  core3/VDD_INT
  ......     ................      ................       .....      ....    .........   ........    .....     ....  ............
  ......     ................      ................       .....      ....    .........   ........    .....     ....  ............
  ......     ................      ................       .....      ....    .........   ........    .....     ....  ............
```


©2020 ANSYS, Inc. / Confidential

EM Metal Reports: Argument(s)

`emir_reports.write_em_metal_report`

Required Argument
electromigration_view

Optional Arguments	
output_file	"./metal.em.rpt"
ignore_nets_file	
columns	
sort	True
sort_order	'descending'
sort_columns	['violation']
formats	
header	

Optional Arguments	
footer	
max_lines	5000
nets	None
nets_regex	None
layers	None
layer_regex	None
em_range	None
ignore_sliver	True

©2020 ANSYS, Inc. / Confidential

EM Metal Reports: columns

`emir_reports.write_em_metal_report`

Valid column names
metal_line_number_factor
layer
applied_duty_ratio
lifetime_factor
violation
vomin
power_grid
em_type

Valid column names
constraint_expr
net
from
constraint
seg_length
current
to
length

Valid column names
si_width
applied_pulse_width
pulse_width
duty_ratio
width
status
current_direction

©2020 ANSYS, Inc. / Confidential

EM Metal Report: Usage Examples

Report all EM Violations in the output report

```
emir_reports.write_em_metal_report(emv, max_lines=None, sort=False)
```


©2020 ANSYS, Inc. / Confidential

EM Metal Report: Usage Examples

Report violations for only one Net in the output file

```
# ignore_nets_file
VDD
core3/VDD_INT
```

```
emir_reports.write_em_metal_report(emv, ignore_nets_file='./ignore_nets_file')
emir_reports.write_em_metal_report(emv, nets=['VSS'])
emir_reports.write_em_metal_report(emv, nets_regex=CMRegex('VSS'))
```


©2020 ANSYS, Inc. / Confidential

EM Metal Report: Usage Examples

Report violations for only two layers in the output file

```
emir_reports.write_em_metal_report(emv, layers=['metal1', 'metal2'])
emir_reports.write_em_metal_report(emv, layers_regex=CMRegex('metal[4,5]'))
```


©2020 ANSYS, Inc. / Confidential

EM Metal Report: Usage Examples

Specify the violation ranges for the output

```
emir_reports.write_em_metal_report(emv, em_range=80)
emir_reports.write_em_metal_report(emv, em_range=(80, 500))
```


©2020 ANSYS, Inc. / Confidential

Electromigration Reports: Via Segments

`emir_reports.write_em_via_report`

```
emir_reports.write_em_via_report(emv)
```

```
# em_type = DC, ignore_sliver = True
# layer       loc_x        loc_y      via_length  via_width  current    constraint violation  status net
#               (u)          (u)          (u)         (u)       (A)        (u)        (%)
   via8      662.145      197.505        0.4         0.4     6.207e-05  0.0001244    49.88     PASS  VDD
   via8      659.625      197.505        0.4         0.4     6.206e-05  0.0001244    49.87     PASS  VDD
   via8      660.465      197.505        0.4         0.4     6.192e-05  0.0001244    49.76     PASS  VDD
   via8      661.305      197.505        0.4         0.4     6.192e-05  0.0001244    49.76     PASS  VDD
   via7      660.885      197.505        0.4         0.4     4.986e-05  0.0001244    40.07     PASS  VDD
   via7      661.725      197.505        0.4         0.4     4.976e-05  0.0001244    39.99     PASS  VDD
   via7      660.045      197.505        0.4         0.4     4.971e-05  0.0001244    39.95     PASS  VDD
   via7      659.205      197.505        0.4         0.4     4.934e-05  0.0001244    39.65     PASS  VDD
   via7      662.565      197.505        0.4         0.4     4.929e-05  0.0001244    39.61     PASS  VDD
   via2      660.885      197.505        0.07        0.07    0.0001287  0.0003871    33.26     PASS  VDD
   via8      762.145       19.705        0.4         0.4     3.397e-05  0.0001244    27.29     PASS  VSS
   ...       .......      .......       ....        ....     .........  ........    .......    ......  ....
   ...       .......      .......       ....        ....     .........  ........    .......    ......  ....
   ...       .......      .......       ....        ....     .........  ........    .......    ......  ....
```


©2020 ANSYS, Inc. / Confidential

EM Via Reports: Argument(s)

`emir_reports.write_em_via_report`

Required Argument
electromigration_view

Optional Arguments	
output_file	"./via.em.rpt"
ignore_nets_file	
columns	
sort	True
sort_order	'descending'
sort_columns	['violation']
formats	
header	

Optional Arguments	
footer	
max_lines	5000
nets	None
nets_regex	None
layers	None
layer_regex	None
em_range	None
ignore_sliver	True

©2020 ANSYS, Inc. / Confidential

EM Via Reports: columns

emir_reports.write_em_via_report

Valid column names
loc_x
loc_y
num_cuts
metal_line_number_factor
layer
from
wb
constraint

Valid column names
violation
lb
current
to
wu
lu
power_grid
via_length

Valid column names
em_type
via_width
net
is_upstream
constraint_expr
lifetime_factor
status
current_direction

©2020 ANSYS, Inc. / Confidential

EM Via Report: Usage Examples

Report all EM Violations in the output report

```
emir_reports.write_em_via_report(emv, max_lines=None, sort=False)
```


©2020 ANSYS, Inc. / Confidential

EM Via Report: Usage Examples

Report violations for only one Net in the output file

```
# ignore_nets_file
VDD
core3/VDD_INT
```

```
emir_reports.write_em_via_report(emv, ignore_nets_file='./ignore_nets_file')
emir_reports.write_em_via_report(emv, nets=['VSS'])
emir_reports.write_em_via_report(emv, nets_regex=CMRegex('VSS'))
```


©2020 ANSYS, Inc. / Confidential

EM Via Report: Usage Examples

Report violations for only two layers in the output file

```
emir_reports.write_em_via_report(emv, layers=['via1', 'via2'])
emir_reports.write_em_via_report(emv, layers_regex=CMRegex('via[4,5]'))
```


©2020 ANSYS, Inc. / Confidential

EM Via Report: Usage Examples

Specify the violation ranges for the output

```
emir_reports.write_em_via_report(emv, em_range=80)
emir_reports.write_em_via_report(emv, em_range=(80, 500))
```


©2020 ANSYS, Inc. / Confidential

Resistance Reports: SPR

`emir_reports.report_instance_pin_spr`

```
emir_reports.report_instance_pin_spr(ev)
```

#pin	spr_value	cell_name	x	y	layer	instance_name
VSS	345.183	INV_X4	1116.895	1.4	metal1	ZINV_142_inst_52481
VSS	343.164	INV_X4	1113.855	1.4	metal1	ZINV_142_inst_52483
VSS	342.405	INV_X4	1112.715	1.4	metal1	ZINV_142_inst_52479
VDD	327.482	INV_X4	1116.895	0.0	metal1	ZINV_142_inst_52481
VDD	325.467	INV_X4	1113.855	0.0	metal1	ZINV_142_inst_52483
VDD	324.712	INV_X4	1112.715	0.0	metal1	ZINV_142_inst_52479
VDD	223.198	INV_X4	1173.535	435.565	metal1	core3/ZINV_221_inst_26118
VDD	213.474	BUF_X8	1167.65	435.565	metal1	core3/ZBUF_49_inst_25959
VDD	211.294	CLKBUF_X1	1121.47	1094.8	metal1	ZBUF_46_inst_55212
VDD	210.916	CLKBUF_X1	1120.9	1094.8	metal1	ZBUF_36_inst_55211
VDD	210.538	CLKBUF_X1	1120.33	1094.8	metal1	ZBUF_45_inst_55210
.....						
.....						
.....						

©2020 ANSYS, Inc. / Confidential

SPR Report: Argument(s)

```
emir_reports.write_instance_pin_spr
```

Required Argument
ev

Optional Arguments	
output_file	“./pin_SPR.rpt”
max_lines	5000
threshold	None
report_type	best
ignore_disconnects	True
ignore_cell_types	[‘is_filler_cell’, ‘is_decap_cell’]

Optional Arguments	
sort	True
columns	
formats	
header	
sort_columns	[‘spr_value’]

©2020 ANSYS, Inc. / Confidential

SPR Report: columns

```
emir_reports.write_instance_pin_spr
```

Valid column names
pin_name
spr_value
cell_name
x
y
layer
instance_name

©2020 ANSYS, Inc. / Confidential

SPR Report: Usage Examples

Custom output columns in the report

```
columns = ['spr_value', 'layer', 'pin_name', 'instance_name']
formats = '{0:10.4} {1:10} {2:10} {3}\n'
header = '{0:10} {1:10} {2:10} {3}\n'.format(*columns)
emir_reports.report_instance_pin_spr(ev, columns=columns, header=header,
                                     formats=formats, max_lines=None, sort=False)
```


©2020 ANSYS, Inc. / Confidential

SPR Report: Usage Examples

Ignore Sequential and Filler cells from the output

Skip instance-pin pairs whose SPR is less than 50 ohms

```
ignore_cell_types = ['is_filler_cell', 'is_seq_cell']
emir_reports.report_instance_pin_spr(ev, ignore_cell_types=ignore_cell_types,
                                     sort=False, max_lines=None, threshold=50)
```


©2020 ANSYS, Inc. / Confidential

Resistance Reports: Effective Resistance

`emir_reports.report_effective_resistance`

```
emir_reports.report_effective_resistance(reff)
```

```
#Reff Report
#instance_name                        pin_name     min          avg          max
xofiller!FILLCELL_X1!x12266400y0       VSS         412.694      412.694      412.694
xofiller!FILLCELL_X1!x12266400y14000   VSS         412.694      412.694      412.694
xofiller!FILLCELL_X1!x12264500y14000   VSS         412.568      412.568      412.568
xofiller!FILLCELL_X1!x12264500y0       VSS         412.568      412.568      412.568
xofiller!FILLCELL_X1!x12262600y14000   VSS         412.442      412.442      412.442
xofiller!FILLCELL_X4!x12256900y0       VSS         412.254      412.254      412.254
xofiller!FILLCELL_X8!x12247400y14000   VSS         411.876      411.876      411.876
xofiller!FILLCELL_X8!x12241700y0       VSS         411.498      411.498      411.498
xofiller!FILLCELL_X16!x12217000y14000  VSS         410.364      410.364      410.364
xofiller!FILLCELL_X16!x12211300y0      VSS         409.987      409.987      409.987
xofiller!FILLCELL_X32!x12156200y14000  VSS         407.342      407.342      407.342
xofiller!FILLCELL_X32!x12150500y0      VSS         406.964      406.964      406.964
xofiller!FILLCELL_X32!x12089700y0      VSS         402.898      402.898      402.898
xofiller!FILLCELL_X32!x12028900y0      VSS         398.805      398.805      398.805
.....
.....
.....
```


©2020 ANSYS, Inc. / Confidential

Effective Resistance Report: Argument(s)

`emir_reports.report_effective_resistance`

Required Argument
`reff`

Optional Arguments	
`file_name`	`"./reff_report.rpt"`
`report_type`	`None`
`max_lines`	`5000`
`sort_columns`	`['max_res']`

©2020 ANSYS, Inc. / Confidential

Effective Resistance Report: Usage Examples

Report the effective resistance for all instances in the design

```
emir_reports.report_effective_resistance(reff, max_lines=None)
```


©2020 ANSYS, Inc. / Confidential

Effective Resistance Report: Usage Examples

Report the top 50 instances in the design with the highest effective resistance

```
emir_reports.report_effective_resistance(reff, max_lines=50, report_type='max_res')
```


©2020 ANSYS, Inc. / Confidential

Documentation

Reporting Utilities in RedHawk-SC

Version: 2019.03.20

Table of Contents


```
>>> help(emir_reports)
help(emir_reports)
FUNCTIONS
get_instance_voltage_data(av, scdb=None, tag_name='instance_voltage_chunked_data', report_
macros=False)
  Return a MapReduce object (ChunkedData) holding a defined set of Instance Arc Voltage Da
ta.

report_effective_resistance(reff, file_name='./reff_report.rpt', report_type=None, max_lin
es=5000, sort_columns=['max_res'])
  Create Reff report file.

report_instance_pin_spr(ev, output_file='pin_SPR.rpt', max_lines=5000, threshold=None, rep
ort_type='best', ignore_disconnects=True, ignore_cell_types=['is_filler_cell', 'is_decap_c
ell'], exclude_cells=None, sort=False, columns=None, formats=None, header=None, sort_colum
ns=['spr_value'], report_pg_arcs=None)
  Write the SPR values at the instance pin to a file.
```

©2020 ANSYS, Inc. / Confidential