Mixed signal ASIC : implementation and validation

Part 3 : assemble design and signoff checks

Bertrand GENNERET, Principal Application Engineer 7 Oct. 2022



cadence

Agenda

- Top level chip assemble
- Static Timing Analysis
- Crosstalk effect analysis
- Power analysis
 - Power consumption
 - Irdrop
 - Electromigration
- DRC / LVS
- Simulation
- Conclusion



Top level chip assemble



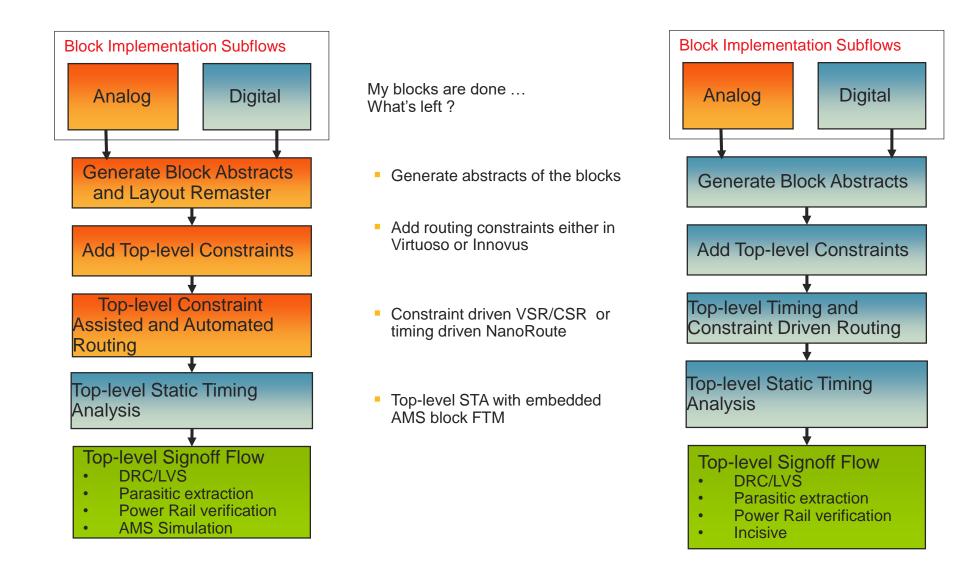
Analog on Top: implementation of top level with digital block

- For the implementation of top level, we will need to create an abstract for the digital block and replace it at the top level. The abstract is a footprint of the layout view and contains only the boundary, pins and blockage information of the block (refer to presentation, part 1).
- After the layout of the digital block is implemented (presentation, part 2) the next step is to create a more accurate abstract with Abstract generator with cover blockages in the block and adding antenna information to the abstract.
- Once the abstract is generated we do replace the initial softAbstract view of the digital block at the top level with the implemented abstract view in Virtuoso.

cādence

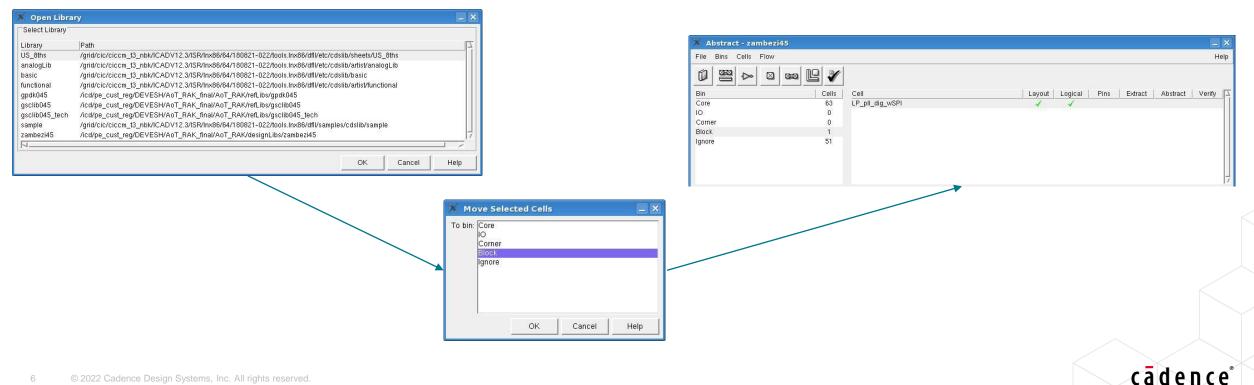
User can then execute top level routing and DRC/LVS flows

Top level assembly flow



Analog on Top : detailed abstract creation

- Launch the abstract generator: abstract –log abstract.log
- Click File > Library > Open
- If necessary, select the core bin, and move the digital cell to bin Block



Extract antenna info

 Change to extract step and in the antenna tab you will see the options to calculate the antenna values.

Step	Signal P	ower Antenna General		
 ◇ Pins ◆ Extract ◇ Abstract 	 Calculate Calculate Calculate Calculate Calculate Calculate 	hierarchical antenna input pin antenna output pin antenna inout pin antenna antenna metal area antenna metal side area nment for Antenna Regions		
	Louor	Geometry Specification	Region	Oxide
	Layer			
	Poly	Poly and Oxide	Gate	

Blockage cover

 Change to the Abstract Step in the form and check the Blockage TAB. The option is to create COVER blockages for all layers and block any routing over the block when the top level is being routed.

ns		ust Block	age Density Fracture Site o	Overlap Grid:	s	
Extract Abstract		Layer	Geometry Specification Blockage		Pin Cutout	Max Spac
		Metal3	Metal3	Cover		
		Metal4	Metal4	Cover		
	Metal5 Metal6 Metal7 Metal8	Metal5	Cover			
		Metal6	Metal6	Cover		
		Metal7	Metal7	Cover		
		Metal8	Metal8	Cover		
	1	Metal9	Metal9	Cover		
		Metal10	Metal10	Cover		
		Metal11	Metal11	Cover		
	1		1			

Use the abstract at top level

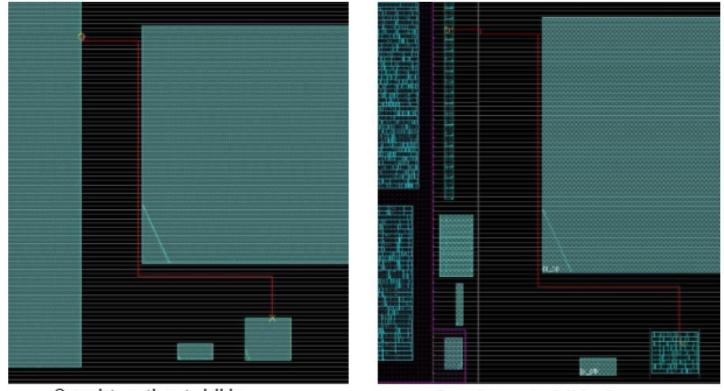
- Open the top level in Virtuoso XL
- Click on tools > remaster instances
- Replace the view name with the abstract created

X Remast	er Instances <@ip-1	72-18-22-22	5>	_ = ×
Search for		Update to		
Library	zambezi45	Library	zambezi45	
CellName	LP_pll_dig_wSPI	Cellname	LP_pll_dig_wSPI	Browse
ViewName	layout_WS	ViewName	abstract	
Check Termi	nals 😧	S		
		0	K <u>C</u> ancel App	ly <u>H</u> elp



Why do we need to flatten the design for analysis?

 In many mixed-signal designs, the digital logic exists at various levels of the physical hierarchy. To perform an accurate analysis of such timing paths, the timer requires the digital logic in the lower levels of the hierarchy to be exposed for creation of the complete timing path. The digital tool should be able to trace the logical and the physical connectivity for the entire timing path.

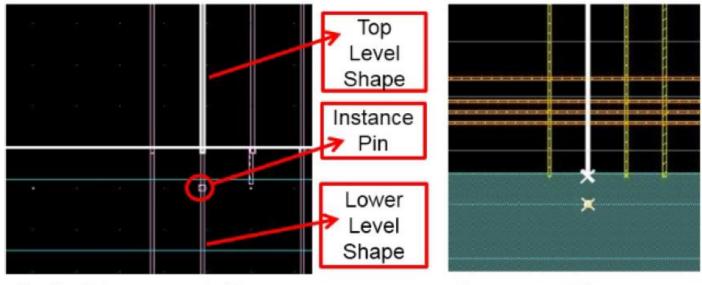


Complete path not visible after initial design load

Complete path visible after assembleDesign

Requirements for Correct Connectivity Propagation

- The nets on the instance terminals should be properly connected for logical connectivity to be propagated. Mosaics should be avoided in the design because although Innovus creates the instance from the mosaic, assemble_design does not map the top-level nets to the lower-level mosaic instances.
- All cells should have shape pins on interconnect layers placed on the cell boundary.



Top level shape connected to instance pin by shape at lower level Innovus sees this as an open

Simplified .lib VS flat design

- Liberate AMS .lib can be generated for the mixed signal blocks such as
 - ADC
 - DAC
 - PLL
 - SERDES
- Flattening the design, to extract Verilog / spef for extraction is another option.
 - Avoid this solution if digital block is too depth in hierarchy

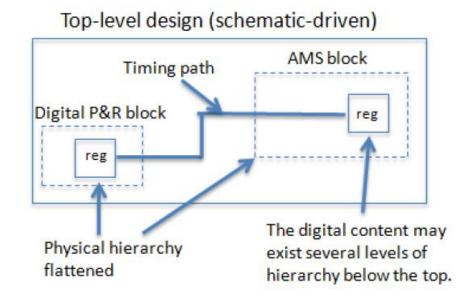


Static Timing Analysis

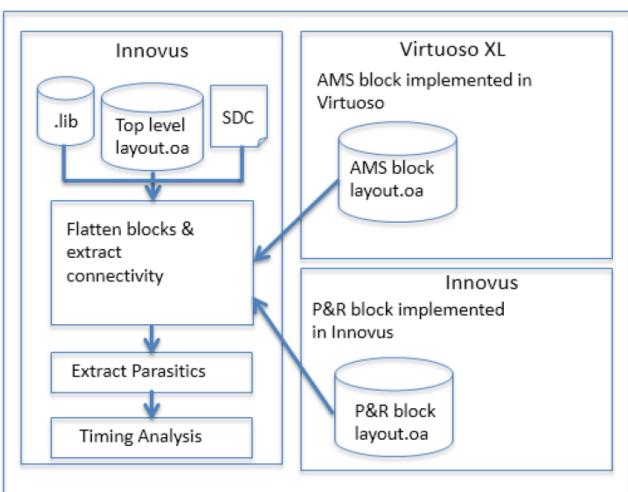


Running STA by Flattening the Design, schematic driven

- In this method, the Innovus command assemble_design is used to flatten the physical hierarchy to bring the instances and wires at the lower physical level to the top for parasitic extraction and timing analysis.
- Note that the physical hierarchy of the design is flattened only to enable static timing analysis. It does not alter the physical structure of the design in any way.



Running STA by Flattening the Design, schematic driven



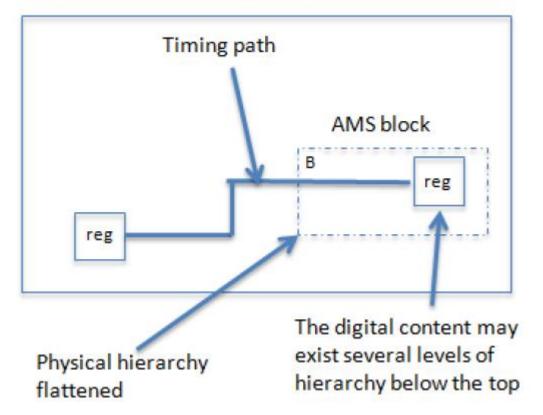
Top-level schematic-driven mixed signal flow

Running STA by Flattening the Design, schematic driven

- If the top-level design is originally a schematic-driven design implemented by Virtuoso, the init_design command should be used to load the OpenAccess design.
- Commands to load the design:
 - o set_db init_power_nets {VDD AVDD}
 - set_db init_ground_nets {GND AGND}
 - o read_mmmc {scripts/viewDefinition.tcl}
 - o read_physical -oa_ref_libs {gsclib045}
 - o read_netlist -oa_cell_view {mylib top layout}
 - init_design

Running STA by Flattening the Design, netlist driven

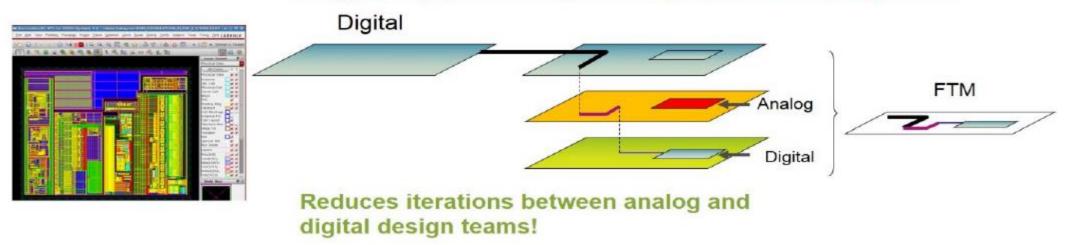
Top-level design (netlist-driven)



17 © 2022 Cadence Design Systems, Inc. All rights reserved.

Full Timing Model

• The Innovus Digital implementation system enables digital paths and logic within mixed sinal hierarchy to be extracted and included for top-level STA.



Easy, comprehensive, and accurate STA using FTM!

Basic FTM flow steps

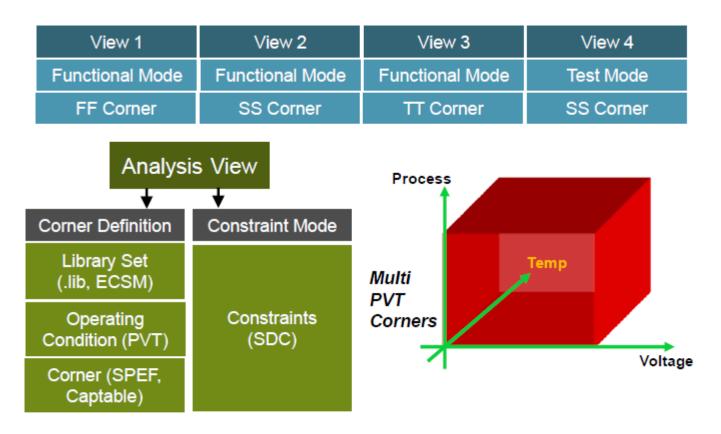
- Load the AMS block as top in Innovus
 - read_db command can be used to retrieve the OpenAccess database
- Use assemble_design to physically flatten the digital logic to top AMS block
- Save the full AMS block Verilog netlist
- Run physical parasitic extraction and generate spef for each RC corner
- Create FTM using the createILMDataDir command for the AMS block
- Load the top level design and specify the FTM path to run the top-level STA

Setup for timing and power analysis

- When running such analysis the job is divided in several steps:
 - Look at the corner to be analysed (user specified) and do pruning to optimize runtime
 - Routing to get wire topology estimations
 - Extraction to get RC network
 - Call timing or power engine to compute results
 - Reporting
- Timing analysis is called regularly during the flow to get a quality of results status.



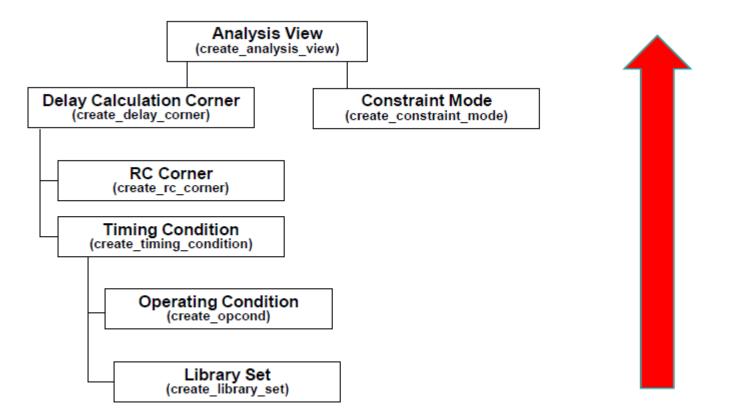
Multi Mode Multi Corner (MMMC) analysis



Concurrent analysis/optimization after views are set up



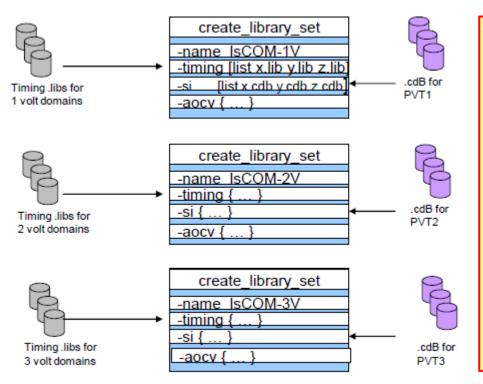
Setting up one analysis view





2 © 2022 Cadence Design Systems, Inc. All rights reserved.

Creating a library set



- Library sets are created so that library references can be created once and conveniently referred to through the specification process.
- The same library set can be referenced multiple times by different delay calculation corners.
- Creates linkage between timing libraries and signalintegrity libraries (.cdb).



Creating a RC corner

- An RC corner object provides the software with all of the information necessary to extract and use the RCs for delay calculation.
- RC corner objects also control the attributes for running gate-level extraction sequentially on each RC corner.
- For each active RC corner in the design, the software extracts and stores a unique set of parasitics. You must use the RC corner scaling attributes when running the software in Multi-Mode Multi-Corner (MMMC) analysis mode.

To create an RC corner run the following:

```
create_rc_corner
```

For example, the following command creates an RC corner called rc-typ that uses the Quantus[™] QRC tech file myTech_nc.qrctech, and derates the resistance values based on the temperature of 50° Celsius. In this example, the –temperature option is used to override the temperature specified in the QRC techfile:

create_rc_corner -name rc-typ -qrc_tech myTech_nc.qrctech -temperature 50



Timing conditions

- A timing condition is a set of libraries at a specific operating condition. Timing conditions are assigned to power domains or supply sets, and effectively describe the delay calculation requirements for the domains.
- A timing condition is associated with a library_set.
- A timing condition is optionally associated with an operating condition (if not specified uses the individual PVT in each library for that given library).
- To create a timing condition run the following:

```
create_timing_condition
```

Example

```
create_timing_condition -name tcWCCOM
-library_set lsCOM-1.0 -opcond_library slow_1.5.lib
-opcond WCCOM_1.5
```



Creating a delay calculation corner

- A delay calculation corner provides all of the information necessary to control delay calculation for a specific analysis view.
- Each corner contains information on the libraries to use, the operating conditions with which the libraries should be accessed, and the RC extraction parameters to use for calculating parasitic data.
- Delay corner objects can be shared by multiple top-level analysis views.
- · Binds power domains to timing conditions with @ syntax.

To create a delay calculation corner, use the following command:

```
create_delay_corner
```

Example

```
create_delay_corner -name DC1 -rc_corner rcMax
```

```
-timing_condition {TC1 pd1@TC2 {pd3 pd4}@TC3}
```

In the example above, TC1, the first element, is the default timing condition that applies to all power domains that are not covered by the pd@TC syntax. The default timing condition is mandatory.

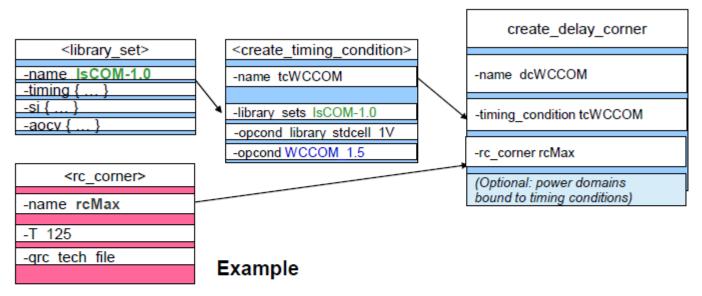
You can specify -early_* and -late_* within a single delay calculation corner to control on-chip variation.

Example

```
create_delay_corner -name_DC1 -timing_condition {TC1 PD2@TC2} -rc_corner QX
```



Delay calculation corners



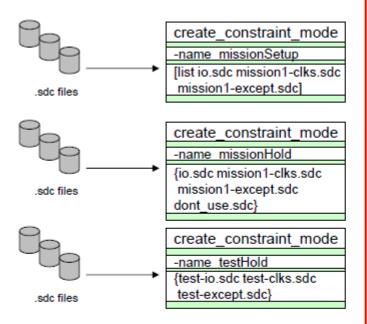
To find out what the delays corners have been set to, enter:

get_delay_corner dcWCCOM -timing_condition tcWCCOM
get_delay_corner dcWCCOM -rc_corner rcMax

Specify the timing condition using the *-timing_condition* and extraction corner with the *-rc_corner* option.



Constraint Modes



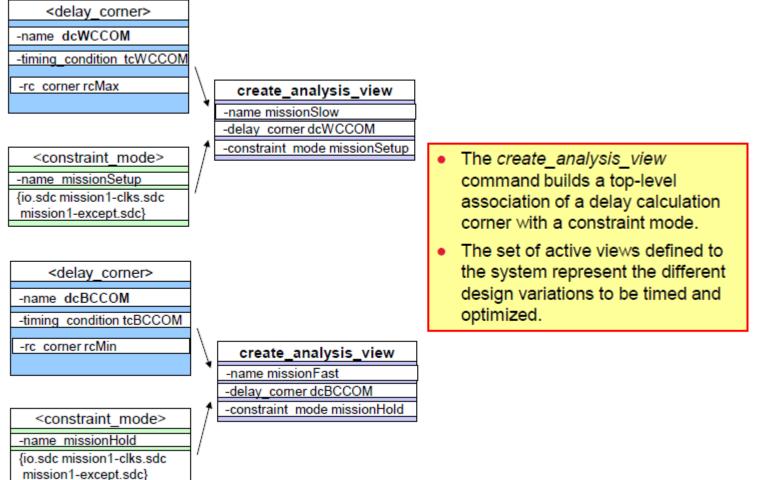
Example

get_constraint_mode missionSetup -sdc_files io.sdc mission1-clks.sdc mission1-except.sdc update_constraint_mode -name missionSetup -sdc_files "io.sdc mission1-clks.sdc mission1-except.sdc fp.sdc" get_constraint_mode missionSetup -sdc_files io.sdc mission1-clks.sdc mission1-except.sdc fp.sdc

- The create_constraint_mode command is used to associate a Tcl list of .sdc files with a named mode.
- SDC files can be shared by many different modes.
- A mode defines one of possibly many different functional, test behaviors, or DVFS modes of a design.
- SDC files contain the clock specifications, conditionalizing constants, I/O timings, and path exceptions that make each mode unique.

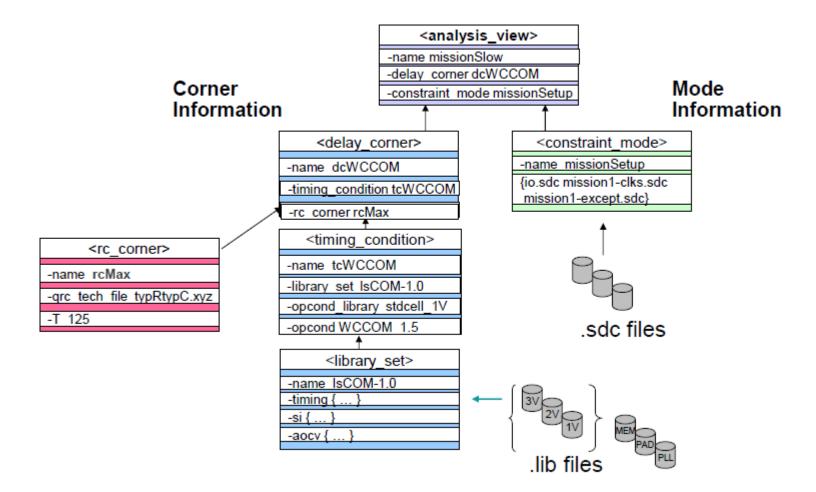


Analysis views





MMMC definition summary





Setting active analysis views

After creating analysis views, you must set which views the software should use for setup and hold analysis and optimization including for leakage and dynamic power optimization.

- These "active" views represent the different design variations that are to be analyzed. Active views
 can be changed throughout the flow to use different subsets of views.
- Libraries and data are loaded into the system as specified by the active views.

To set active analysis views, use the following command:

```
set_analysis_view
```

The following command sets *missionSlow* and *mission2Slow* as the active views for setup analysis, and *missionFast* and *testFast* as the active views for hold analysis:

```
set_analysis_view -setup {missionSlow mission2Slow} \
-hold {missionFast testFast} -leakage {max_leakage_view} -dynamic
{max_dynamic_view}
```



Checking MMMC configuration

To report your current multi-mode multicorner configuration, run the following commands:

report_analysis_views

all_analysis_views

reports a list of all views

```
get_db analysis_views -if
  {.is_setup == "true"}
```

```
get_db analysis_views -if
  {.is_hold == "true"}
```

```
get_db analysis_views -if
  {.delay_corner.name ==
   "dcWCCOM"}
```

```
+ ALL Views
+ Analysis View: default_analysis_view_setup
+ Delay Calc Corner: default_delay_corner_max
+ timing_condition: default_delay_corner_max
+ timing_condition: default_libset_max
+ timing: /home/vimita/InnovusBlk_CUI_17_1/FPR/work/place.inn/libs/mmmc/rps73gwc-lite_slew.lib
/ home/vimita/InnovusBlk_CUI_17_L/FPR/work/place.inn/libs/mmmc/rps73gwc-lite_slew.lib
/ home/vimita/InnovusBlk_CUI_17_L/FPR/work/place.inn/libs/mmmc/rplace.lib/
/ home/vimita/InnovusBlk_CUI_17_L/FPR/work/place.inn/libs/mmmc/slew.lib
/ home/vimita/InnovusBlk_CUI_17_L/FPR/work/place.inn/libs/mmmc/slew.lib
/ home/vimita/InnovusBlk_CUI_17_L/FPR/work/place.inn/libs/mmmc/slew.lib
/ home/vimita/InnovusBlk_CUI_17_L/FPR/work/place.inn/libs/mmmc/slew.lib
/ home/vimita/InnovusBlk_CUI_17_L/FPR/work/place.inn/libs/mmmc/slew.cdb
```

Customize the report to show *only* one of the following:

- The active setup or hold analysis views.
- All of the active views.
- All of the defined views in the design, including those that are currently inactive.



cādence

...

Specifying extraction mode

set_db extract_rc_effort_level {low | medium | high | signoff}

Where

- Low enables the detailed extraction engine.
- Medium enables the pre-route extraction engine for preRoute and tQuantus extraction engine for post_route extraction.
- High enables the integrated Quantus extraction engine.
- Signoff enables standalone Quantus extraction.



Running extraction

Timing – Extract RC

	Extract RC	- 🗆 🗙
- Save RC -		
🗌 Save Seti	oad to DTMF_CHIP.setload	6
🔲 Save Set	Resistance to DTMF_CHIP.set	res 🕑
Save SPF	to DTMF_CHIP.spf	B
🗾 Save SPE	F to DTMF_CHIP.spef	Ð
RC Corner to	Output default_rc_comer_wor	st 🗖

Command
extract_rc
To delete the extracted
parasitics but to maintain the RC
extraction modes that were set,
run the command
reset_parasitics

- Setload format (lumped capacitance on nets)
- Setres format (resistance on nets)
- SPF saves the DSPF (detailed standard parasitic format) file
- SPEF (standard parasitic exchange format)



Performing timing analysis

The *time_design* command reports the results from each active analysis view, as well as an aggregated summary.

Syntax

time_design

By default, *time_design* creates only an aggregated summary.

Use the *-expanded_views* option to report timing of all views.

For multi-mode multi-corner analysis, the software creates a separate directory for each view.

For example, if your design has two analysis views, view1 and view2, the output reports are generated in the ./timingReports/view1 and ./timingReports/view2 directories.

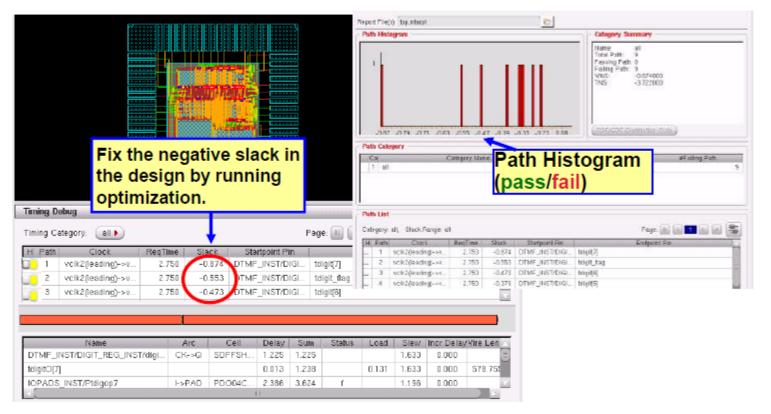
	Command				
time_	design		<pre>-expanded_views</pre>		

Setup mode	all	reg2reg	default
WNS (ns): TNS (ns): Violating Paths: All Paths:	-4.041 22	-0.545 -4.041 22 580	2.895 0.000 0 266
AV_HL_FUNC_MAX_RC1	-0.545	-0.545	2.895
	-4.041	-4.041	0.000
	22	22	0
	622	580	266
AV_HL_FUNC_MAX_RC2	0.019	0.019	3.815
	0.000	0.000	0.000
	0	0	0
	622	580	266
AV_LO_FUNC_MAX_RC1	7.455	7.455	10.895
	0.000	0.000	0.000
	0	0	0
	622	580	266



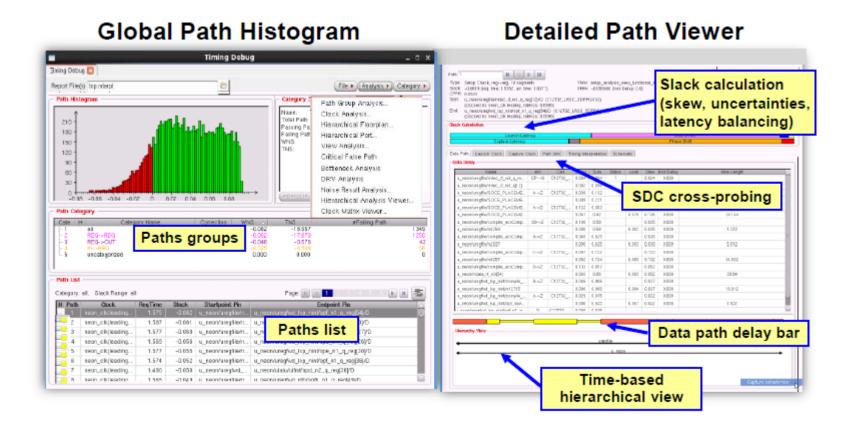
Timing debug utility

After running timing analysis, choose **Timing – Debug Timing** to help debug the causes of timing violations in your design.





Timing debug utility



Reduces thousands of failing paths to a small number of unique problems to solve.



Examples of debugging with Timing Path Analyzer

Launch and capture latency components are not aligned. Therefore, there can be large clock latency mismatch in this path.

and a local division of the local division o						
8		aunch Latency	100011110	Data Delay		
1000	Capture Latency	Phase Shift		Negative Slack	Setup:	0,185 (3,02)

The cycle adjustment bar in the required time indicates the presence of multicycle path.

Launch Latency	Data De	lay
Capture Latency	Phase Shift	Cycle Adjustment

Large input delay in an I/O path is represented by the light-blue bar in the arrival time.

Slack Calculation				
Network Insertion I	Delay	Input Delay		
Source Insertion Delay	Capture Latency	Phase Shift	Plata Delay: 0.51	



Timing Path Analyzer: Path sdc

The Path SDC tab shows all the SDC constraints that match the topology of the current path. Use this software to debug typical constraint issues.

	Timing Path Analyzer	
	Timing Path Analyzer #	
	Path: 4 N A F M Type: Late External Delay Assertion; regi-out, 6 segments View: default_amalysis_view_setup Stack: 0.0000 (reg, time: 2.7500, arr. time: 2.4440) Skew: 0.0000 (Incr Delay: 0.0) CPPR: 0.0000 CPPR Common Pelnt: Start: DTMF_INST/DKitT_IEQ, INST/digt_out_reg_3/Q (SDFFSHQX1) (clocked by vck2 feading, latency: 0.0000) End: 1dtgt[0] (clocked by vck1 leading, latency: 0.0000)	×
	Slack Calculation	
	Data Delay	
	Phase 5bft	
umber	SDC File: /home/vinita/InnovusBik_18_1/FR/work/noiteCstracted.inv.dat/lins/mmc/dtaf.sdc 8 create_clock [get_pins [DIMF_INST/TEST_CONTROL_INST/i_150/Y0] - name velki - period 7 -waveform [0.3.5] 10 create_clock [get_pins [DIMF_INST/TEST_CONTROL_INST/i_156/Y DIMF_INST/EST_CONTROL_INST/i_154/Y DIMF_INST/TEST_CONTROL_INST/i_156/Y DIMF_INST/I_156/Y DIMF_	



Export sdf file

- Sdf stands for Standard Delay Format
- You can export it from Innovus with the command:
 - write_sdf -edges check_edge -version 3.0 -precision 3 -target_application verilog

•	ut or a rectional port	0->1	transition	0-:	>Z transition	1-:	>Z transition
(IC	PATH i3 o1	0.0) (2:4:5)	(4:	5:6) (2:4:5)(4:5:6))
	Output or a bidirectional p		1->0 transit	ion	Z->1 transitio	on	Z->0 transition
							elay in following format :typical:max

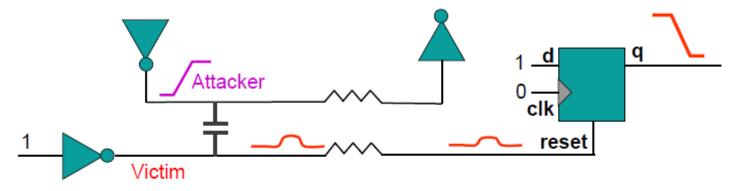


Crosstalk effect analysis

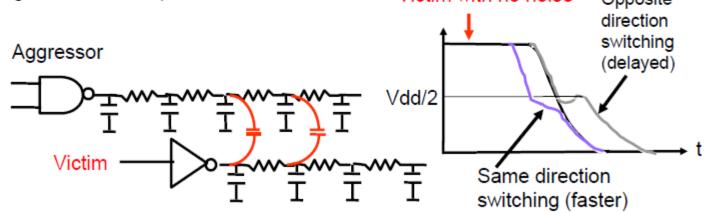


Impact of noise

A glitch noise that exceeds a threshold can cause functional failures.

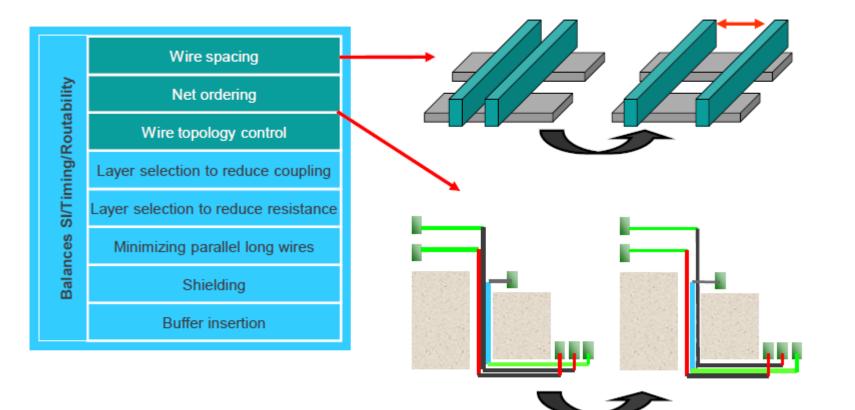


Crosstalk noise can create timing problems involving setup (switching opposite direction) and hold (switching same direction). Victim with no noise Opposite





Crosstalk avoidance

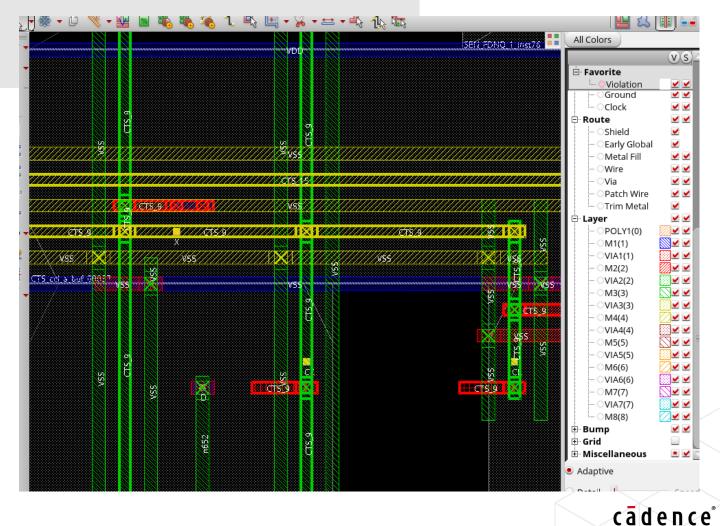


Shielding of clock nets

create_route_type -name clock_route -top_preferred_layer M5 -bottom_preferred_layer M3 -shield_net VSS set_db cts_route_type_top clock_route

set_db cts_route_type_trunk clock_route
set_db cts_route_type_leaf clock_route

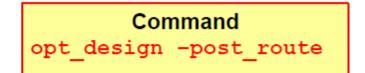
ccopt_design



Fixing crosstalk Post-route

Choose ECO – Optimize Design. Select Post-Route, Include SI, and click SI Options.

Optimizat	tion		1227	o x
C Post-CTS		Post-Route	L	
e				
	🗌 Hold			
alations				
Options)				
ly <u>M</u> ode	Defau	lt Close)		elp
	Post-CTS e plations	e	Post-CTS Post-Route Hold	Post-CTS Post-Route Hold Datations



By default, delay due to Signal Integrity are computed, and glitch fixing is activated if using –postroute.

To compute only base delay you can setup: set_db delaycal_enable_si false

To turn on Signal Integrity delay but turn off glitch optimization you can use: set_db opt_post_route_fix_glitch false

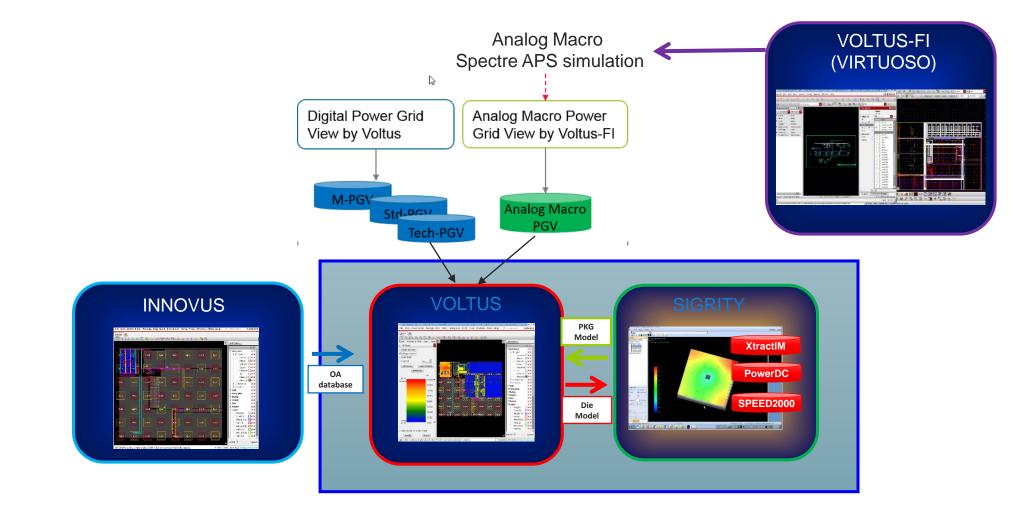


Power analysis

46 © 2022 Cadence Design Systems, Inc. All rights reserved.



Power analysis of Mixed Signal design (Digital on Top)



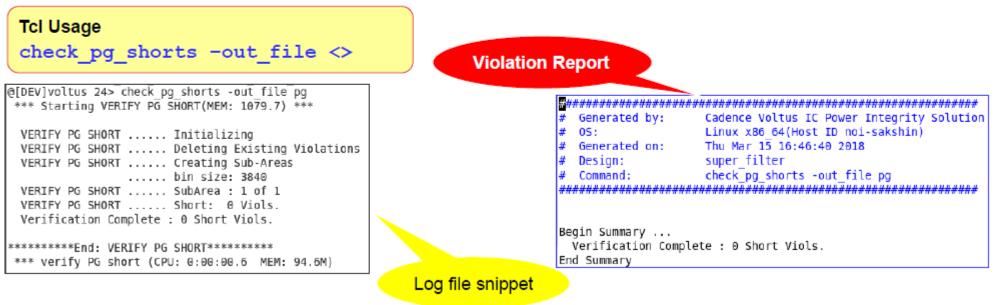


Verifying power ground shorts

It checks for power and ground shorts between two geometries belonging to different nets.

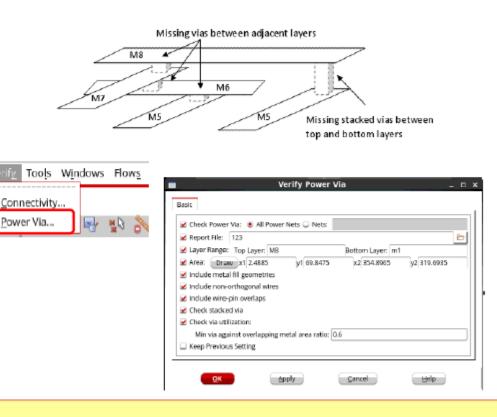
The command performs power and ground short check between the following:

- PG and PG nets
- PG and signal nets
- PG and other special nets



Verifying power via

- Checks for metal geometry overlap on PG nets
- Ignores metal fill geometries by default
- Generates text report with location of missing via on PG net
- Highlights violations in the layout canvas
- Supports stacked via check using the -stacked_via option which checks for missing vias between all non-adjacent as well as adjacent layers.
- Performs missing power via analysis in the user-defined region
- Checks missing power vias on wire and pin overlaps using the -check_wire_pin_overlap option
- Checks via density utilization in metal overlaps



Tcl Command check_power_vias -layer_range {M8 M1} -area {4.805 1.2785} {353.46 256.396} \ -check_fill -non_orthogonal_check -check_wire_pin_overlap -stacked_via

Calculating Power consumption Data

Switching power

- The power consumed by charging and discharging of load capacitance.
- Formula: P_{avg} = ½*C_{load} * V² * F* A

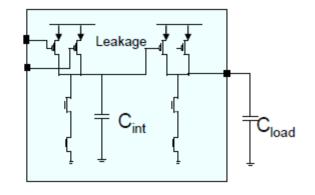
where C_{Load} is the output capacitive loading, V is the voltage, F is frequency, and A is the average switching activity either from VCD or computed.

Internal power

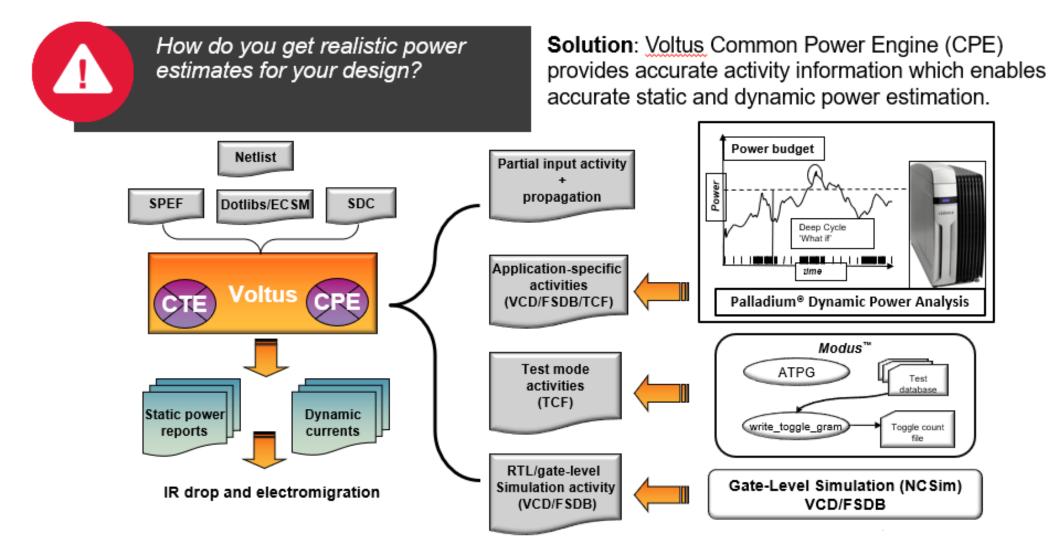
- The power consumed in charging and discharging of interconnect and device capacitances internal to a cell:
 - Input-pin-related internal power
 - Arc-based internal power
 - Crossbar current

Leakage power

- The power consumed by devices when they are not switching
- Supports state-dependent leakage power calculation based on .lib file
- Requires state-dependent leakage library characterization
- Supports K factor (for process, temperature, and voltage) in the .lib file
- High-leakage power for 130nm; significant issue for 90nm and below



Activity setup for power calculation



Tcl command sample: static (average) power calculation

read_db innovus.db

```
read_spef ../innovus_cui/test_top_cbest.spef -rc_corner cbest
read_spef ../innovus_cui/test_top_cworst.spef -rc_corner cworst
```

```
set_analysis_view -setup {func_worstlib_cworst} -hold {func_worstlib_cworst}
```

set_default_switching_activity -reset
set_default_switching_activity -input_activity 0.2 -sequential_activity 0.2

#read_activity_file -format VCD -start 10300ns -end 10500ns funcmode3.vcd

set_db power_method static set_db power_report_statistics true set_db power_write_db true set_db power_write_static_currents true Generate db to be able to display power maps Generate current files (.ptiavg) to be used later for irdrop

set_power_output_dir static_power_cworst
report_power -out_file static_power_cworst/power.rpt



Power reports

After the static-power database is generated, you can generate incremental static power reports based on:

Group

Macro

Total

Sequential

Combinational

Clock (Combinational)

....................

Clock (Sequential)

Power

2.365

0.6282

0.3023

0.01212

3.307

3.87233e-11 F Total instances in design: 2674 Total instances in design with no power: Total instances in design with no activity: Total Fillers and Decap:

..........

0.2023

0.3563

0.1952

1.256

Highest Leakage Power: ring/clk FENCE MSV Fence I 10 (CLKBUFX20): 0.000357

0.002453

0.5

.........

0.06186

0.01079

0.001092

.

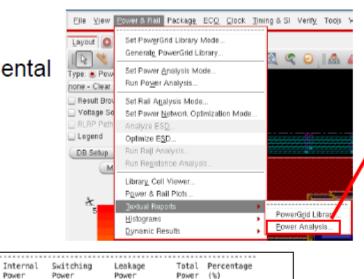
0.2288

8.155

Power Distribution Summary: Highest Average Power:

Total Cap:

- Clock domain
- Power domain •
- Hierarchy level •
- Instance and cell
- Cell type ٠
- Net based
- MMMC view based



.........

0.5

2.722

1.046

4.792

0.5083

0.01566

.

56.8

10.43 ē

21.83

10.61

0.3268

......

160

0.5

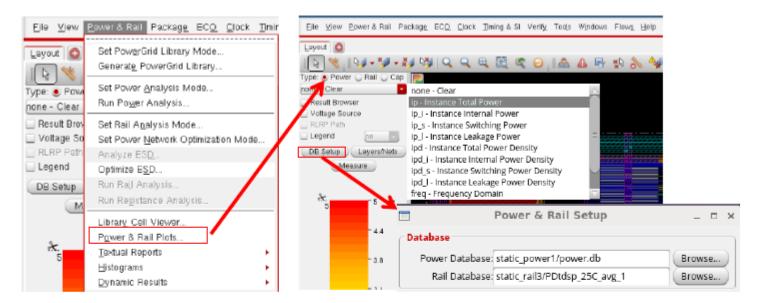


Tcl Command

report power -out file clk.rpt -clock network pll clk -cell type all -sort total

u0 (pll):

Viewing power results maps



If the power database is not specified, the static power stored in memory is used to plot the data.

```
TCL Command
read_power_db -in_file power.db
gui_set_power_rail_display -plot <ip | ip_i | ip_s | ip_l | freq | td |
slack |load> \ -enable_result_browser true
```



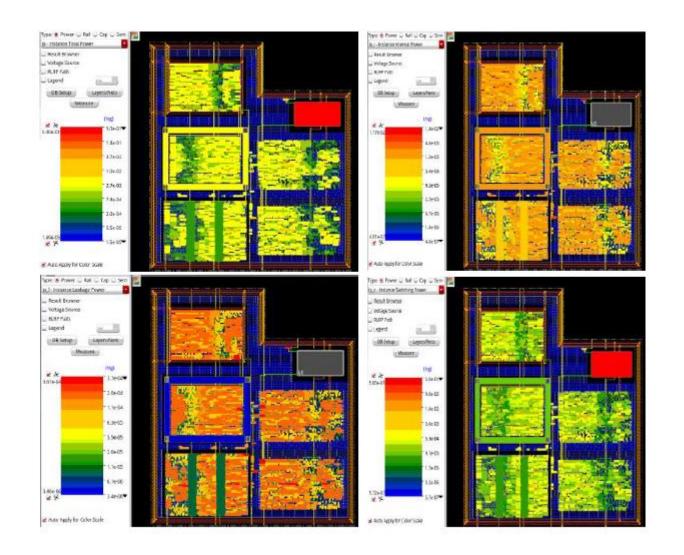
Available power maps

Static power can be read in one of two ways:

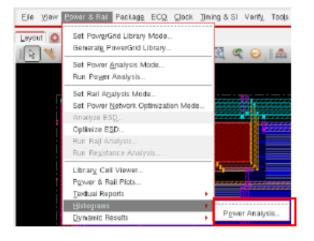
- From calculated power stored in memory during the session
- From *power.db* generated during static power calculation

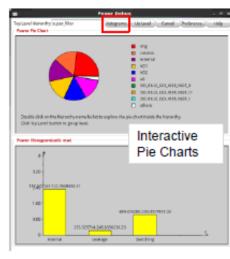
You can display the following plots:

- Instance total power and density
- Instance internal power and density
- Instance switching power and density
- Instance leakage power and density
- Frequency domain
- Slack
- Transition density
- Loading capacitance



Histograms





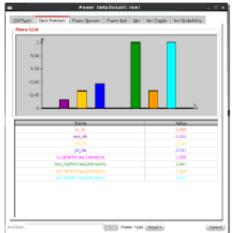
Histogram by:

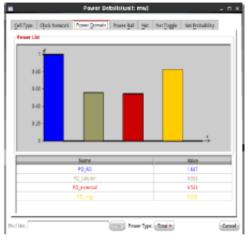
- Cell types
- Power domain
- Power nets
- Clock domain
- Net toggle
- Net probability





Power DetailsOunit: mwb





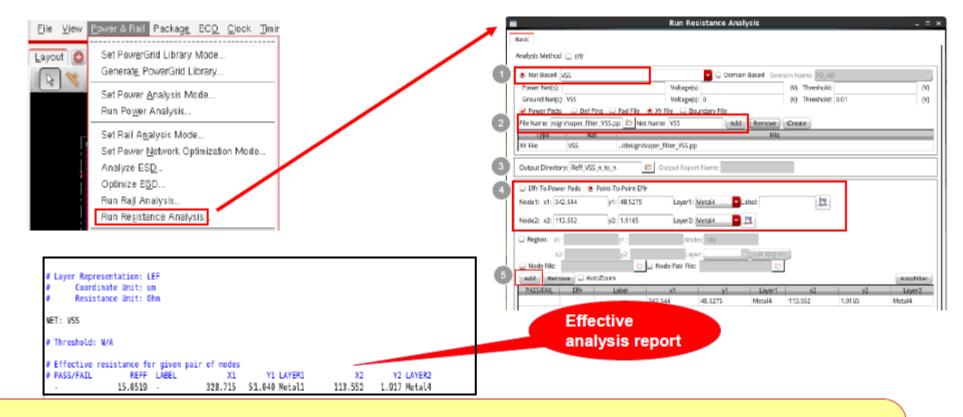
Debugging instance power

Instance: TDSP_CORE_INST0/MPY_32_INST/M16X16_INST/mul_8_14/g5477

report_inst_power TDSP_CORE_INST0/MPY_32_INST/M16X16_INST/mul_8_14/g5477 -out_file pwr.rpt

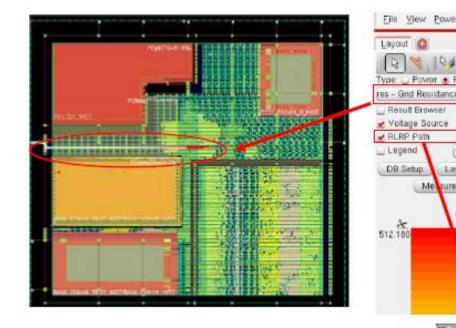
Instance: TDSP CORE INST0/NPV 32 INST/M16X16 INST/mul 8 14/g5477	
Cell: ADDFXLMTR	
Liberty file: /home/sakshin/voltus/voltus labs/LIB5/LIB/scmetro cmos10lp rvt ff 1p1v m40c.lib	
Internal power: 0.00175445mW	
Switching power: 0.00062103mW	
Leakage power: 0.00000013nW;	
Total power: 0.00237560mW;	
Direction with Duty Density (confet) Directory(con) [coll clay(con) Dever(ch() Met	
Direction Vortage(V) Duty Density Cap(pf) Rise slew(ns) Fall slew(ns) Power(mW) Net Pin	
Input 1.10000002 0.23330000 2.500000e+08 0.00125000 0.04039999 0.03530000 0 TDSP CORE INST/MPY 32 INST/M16X16 INST/mul 8 14/n 671 A	
Input 1.10000002 0.49840000 1.965091e+08 0.00121900 0.04450000 0.04030000 0 TDSP_CORE_INST/MI6X16_INST/MI6X16_INST/MUL 8_14/n_640 B	
Input 1.10000002 0.25999999 1.804418e+08 0.00144200 0.04290000 0.03810000 0 TDSP_CORE_INST0/MPY_32_INST/M16X16_INST/mul_8_14/n_619 CI	
Output 1.10000002 0.49959999 2.500000e+08 0.00285600 0.18449998 0.12949999 0.00043197 TDSP_CORE_INST0/MPY_32_INST/ab_result[12] S	
Output 1.10000002 0.24609999 2.500000e+08 0.00125000 0.04929999 0.04210000 0.00018906 TDSP_CORE_INST0/MPY_32_INST/M16X16_INST/mul_8_14/n_673 CO	
Leakage power	
When Duty Power	
(((A) & (B)) & (CI)) 0.6302319 0.0302319*1.14889e-10	
(((A) & (B)) & (!(CI))) 0.0860448 0.0860448*1.24213e-10	
({(A) & {!(B})) & (CI)} 0.0304261 0.0394261*1.17852e-10	
(((A) & (!(B))) & (!(CI))) 0.0865972 0.0865972*1.10255e-10	

Resistance analysis



```
Tcl Command
report_resistance \
    -net VSS -output_dir ./Reff_VSS_n_to_n \
    -node_pair_list {{386.0615 110.619 Metal1 536.115 309.011 Metal 4 }}
```

Least-Resistance Path analysis

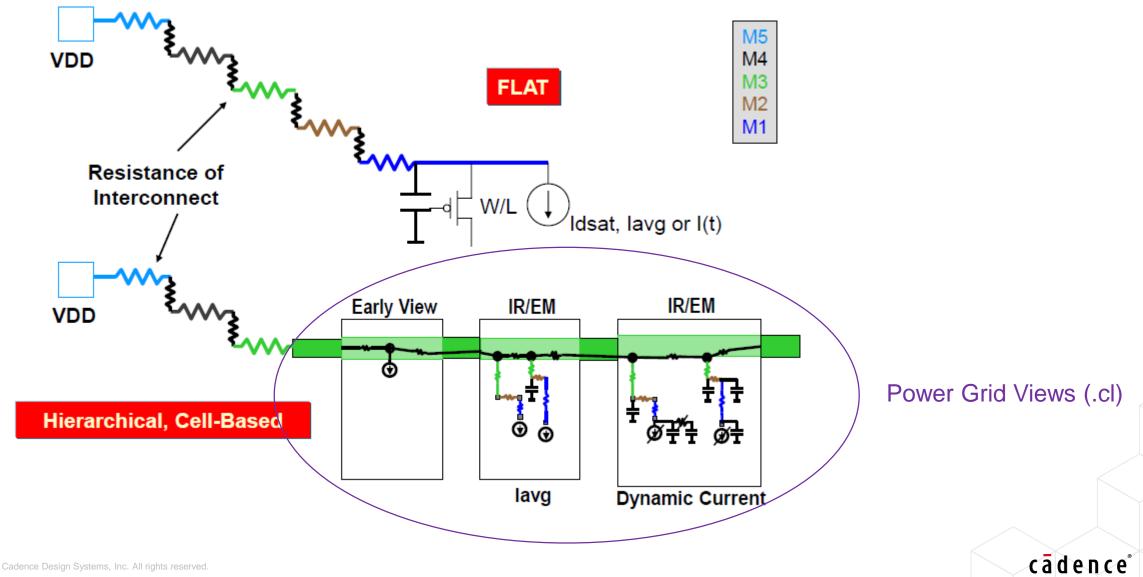


- Similar to *rlrp* analysis, resistance analysis can help to identify weak power grid
- Node-based and grid-based resistance plots
- Interactive trace of least-resistance path in form
- Detailed text report for all segments in LRP

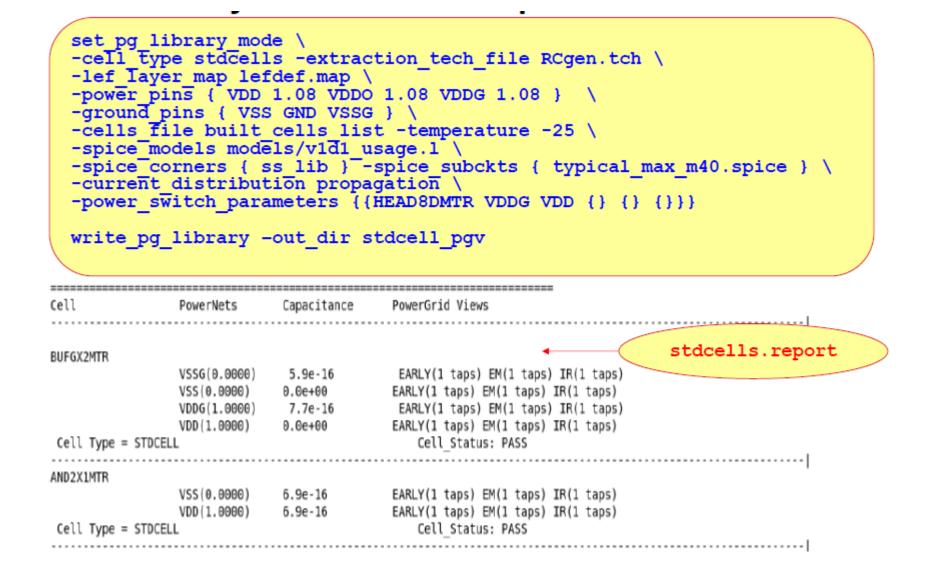
	Text report in Voltus console	
		Tetainel
	le gat ylew Jernese Tags Help	
1	ver/VLa Resistance(0hm) Voltage Drop(V) # Cumulative(Dhm) Vdrop Cumulati	Ine (V)
	4.3164e-81 2.2005e-84 4.3164e-81 2.2005e-84	
	1.6667#-02 1.7883e-06 4.4833e-01 2.3174e-04	
	4.6074e-64 1.1921e-07 4.4877e-00 2.3186e-04	
	1.2560e-02 1.3113e-06 4.6127e-01 2.3317e-04	
	2.3496e-02 8.5593e-85 6.9022e-81 3.1877e-84	
	1.6567e-02 1.1923e-05 7.1209e-01 3.1596e-04	
- 0	8.55844-64 1.1921a-67 7.1374a-61 3.2968a-64	
1.0	1.2588e-82 1.8729e-86 7.2024e-80 3.2115e-84	
	1.3805e+00 3.3555e-04 1.9068e+60 6.3676e-64	
	1.65674-82 7.15256-87 1.92356+88 6.37418-84	
- 0	8.5584e-64 0.8085e+68 1.0243e+68 6.3741e-64	
	1.2508g-02 9.5367e-07 1.9268e+00 6.3037e-04	
	1.3807e+00 1.5925e-04 3.3175e+00 7.8763e-04	
	1.6567e-02 1.5497e-05 3.1342e+40 7.9918e-64	
- 38	3.7375e-00 1.6470e-04 3.5088e+00 9.0386e-04	
	2.5888#-82 2.8268#-86 3.5338c+88 9.8589#-84	
	8.5668e-61 1.1480e-04 4.3897e+68 1.8268e-63	
	2.3000e-02 5.3644e-05 4.4147e+00 1.6202e-03	
- 28	7.3558e-02 1.2279e-05 4.4983e+00 1.8584e-83	
	3.1258e-62 5.1266e-86 4.5195e+68 1.8436e-83	
	1.5317e+01 2.5301e-03 1.0036e+01 3.5737e-03	
1	1.2508e-01 3.3379e-06 1.9963e+01 3.5778e-03	
	2.2373e+60 5.8638e-64 4.2334e+60 4.0534e-63	
1	5.0000e-01 1.3113e-05 4.2034e+01 4.1765e-03	
	0 7578a.81 2 #190a.85 # 3768aa81 # 3987a.83	

Resistance Pa			Enable grid	resist	ance path
Nodel	Rott	S	Enable grie		arroe parri
21 760	14	0.00170448	1005.44,353.10	M7	21621
21759	13,99	0.00178445	1008.74,389.18	M7	21620
21756	13.99	0.00176439	1007.04,369.10	M7	21619
21757	13.85	0.0017825	1015.96,369.18	M7	21518
21758	13.86	0.00178243	1018.26,369.18	M7	21617
4		25		1.44	
Auto Zoon					
1440 2001					
	60	pty	<u></u>	lose	

Ir drop analysis



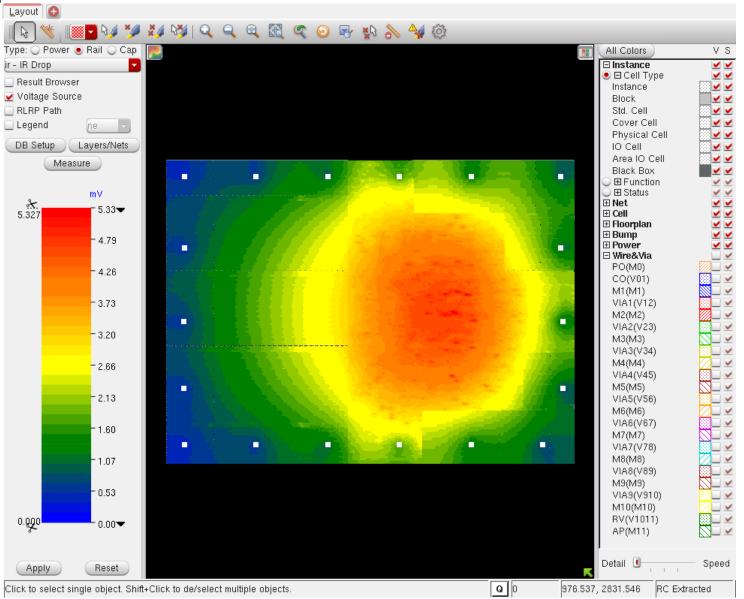
Step1: generate power grid view (PGV = .cl file)



Step2: Rail analysis

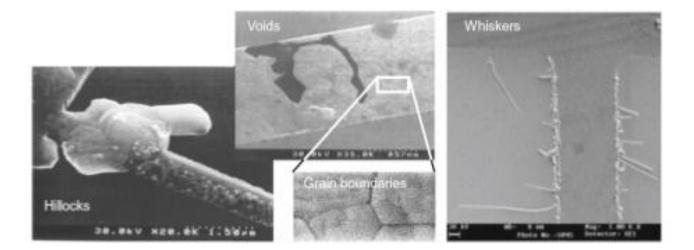


Ir drop map



Electromigration

 If enough electrons collide with a metal lattice ion over a period of time, this can move in the direction of the electron flow, causing damage to the conductor, e.g. voids or hillocks



cādence

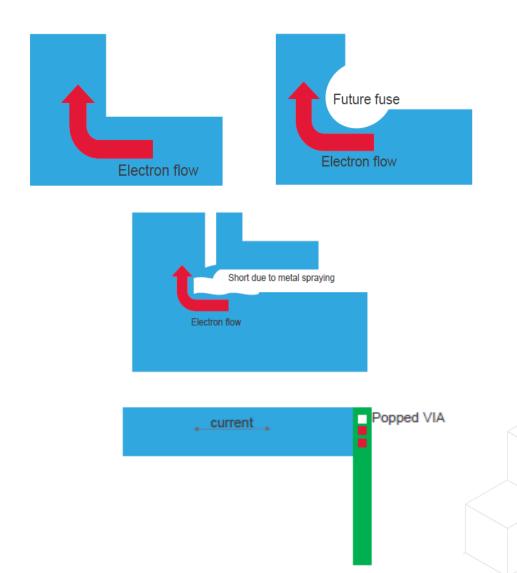
 With the scaling down of technology dimensions, power has not been scaled down proportionally, leading to high currents flowing through smaller and smaller interconnections, thus eventually causing the current densities to exceed the maximum allowed

Consequences of Electromigration

- The following failure mechanisms can occur:
 - If enough atoms move, the wire breaks and becomes an **open circuit**

 If enough atoms move to the same location, a short to an adjacent metal wire is created

 Joule Heating causing the via expansion in the dielectric layer can eventually end up in the popping of the via



cadence

Electromigration rules

Metal wiring level / interlevel connection	Via landing on Bottom metal	I _{max} (mA)
M1		0.95 x (w x 0.9 – 0.002)
Mx (1x metal)		0.92 x (w x 0.9 – 0.002)
My (2x metal)		1.8 x (w x 0.9 – 0.013)
Vx (drawn size 0.05 x 0.05 um2)	Landing on M1/Mx	0.04 per via
Vx rectangular via (drawn size 0.05 x 0.13 um2)	Landing on M1/Mx	2 x 0.04 per via

Temperature	105C	110C	115C	120C	125C
Rating factor of	1.4	1.0	0.9	0.7	0.4
I _{max}					

How to load EM Rules in Voltus?

Below are the different methods sorted by priority

- Use of qrcTechFile with embedded EM rules given by foundry
- Use of ICT EM rules
- Use of iRCX EM rules
- Use of Voltus native format EM rules

Check and update qrcTechFile



 Techgen (under Quantus[™] QRC) is used to update the *qrcTechFile* with the EM model information.

- To update the *qrcTechFile* with the EM model information, use these commands in Quantus:
 - To dump ICT file from *qrcTechFile*:

Techgen -process_out <qrcTechFile>

• To dump info on encrypted qrcTechFile:

Techgen -tech_info <qrcTechFile>

• To include EM rules from .*ICT* into *qrcTechFile*:

Techgen -update_em <qrcTechFile> \

ict_file_with_EM models



EM map

File <u>V</u> iew <u>P</u> ower Rail Package EC <u>O</u> <u>C</u> lock <u>T</u> iming SI Verify Too <u>l</u> s Windows Flow <u>s</u> <u>H</u> elp		🭳 online help	cādence
Layout 💿			
🕟 💘 🔍 🔍 🍳 🕰 🔍 🥑 😼 + 🎾 + 🌮 💖 🔝 🛆 🗟 😵 🌭 🐓 🅸 👘			
ype: 🔾 Power 💿 Rail 🔾 Cap 🔾 Sem 🔜 st193 🛄 inst185 🐘 inst1025 🐘 inst899		III Colors	
i - lavg/llimit	🔀 🕢 Attribute Viewer – euvo	 017 <@euvclo17>	
Versuit Browser	Object Type: Resistor		00
Percentage Range 158 inst188 inst1028 inst896		/alue	Type
Legend ne i inst190 inst189 i inst895 i			Туре
RP Path Trace Clear	Layer	metal2	String
DB Setup Layers/Nets 024 mst1033 mst1031 mst1030	Original Resistance	0.615276 Ohm	Double
Measure 155 inst891 inst892 inst893	Resistance	0.615276 Ohm	Double
. t54	Inductance	N/A	Double
🗹 📩	Current	6.44292e-05 A	Double
0.875 . 150	Current Direction	left	String
0.75 1 inst52 inst885 1 inst457 1 inst451	J/Jmax	11.7167	Double
inst51 inst884 inst460	Jrms/Jrms_max	NA	Double
0.625 .1. Inst461 .1459Inst450Inst449	Node Id	N16888, N16904	Double
0.5	Node Reff	22.66270hm, 22.8625 0hm	Double
0.375	Node Voltage	0.0107975, 0.0108372 V	Double
- 0.251 Inst48. Inst463	Node Voltage Drop/Bounce	10.7975, 10.8372 mV	Double
	Close	Help	
o nst464			
✓ [∞] [−] 0 [−] a buf_00022 inst1071 inst1072 inst1077		OM6(6) OVIA6(6)	
Auto Apply for Color Scale		OM7(7) OVIA7(7)	
Auto Apply for Color Scale Apply Basic Reset Inst466 inst1069 inst1067 inst1078			
Apply Basic Reset 1469		C Detail	Speed
Resistor: R70325, Value: 0.615276 Ohm		433.15150, 314.01350 Sel: 1 Tin	ning Analyzed

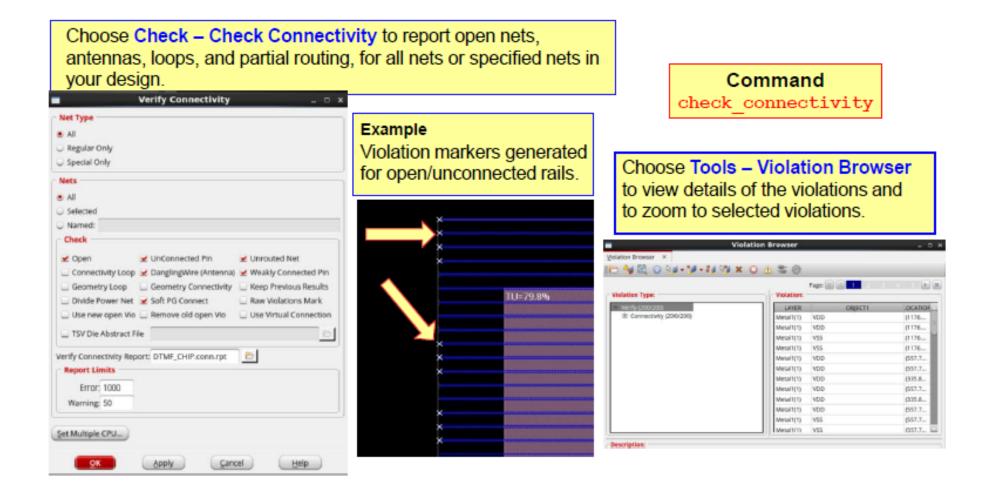


cādence°

DRC / LVS

70 © 2022 Cadence Design Systems, Inc. All rights reserved.

Early checks with Innovus: Verify connectivity



Early checks with Innovus: verify DRC rules

Set the check_* attributes and run the check_drc command to verify DRC rules.

Syntax

```
set_db check_drc_ndr_spacing {true | false}
set db check drc check only {all | regular | special}
```

check_drc [-help] [-area {x1 y1 x2 y2}] [-view_window]
[-check_ndr_spacing] [-check_only {all | regular | special | cell}]
[-layer range <string>] [-limit <integer>] [-out file <string>]

Signoff Verifications with Pegasus

Pegasus features:

- DRC (Design Rule Check)
- XOR (Sometimes referred to as LVL Layout Versus Layout)

- FastXOR (Fast XOR)
- ERC (Electrical Rule Check)
- PERC (Programmable ERC)
- LVS (Layout Versus Schematic)
- SVS (Schematic Versus Schematic)
- Device Signature Generator
- Sign-Off Metal Fill
- Layer Viewer

Running PVS from the command line



pvs -[flow] {options} rules_file1 ... rules_fileN

Example

```
pvs -drc 65nmdrc.rul
pvs -lvs -ui_data -qrc_data 65nmlvs.rul
pvs -lvs -qrc_data 65nmlvs.rul optional.rul
pvs -lvs -gds test.gds -top_cell test -source_verilog top.v \
    -source_cdl stdcells.cdl \
    -source_top_cell test_sch 65nmlvs.rul \
    pvl_edtext.rul
```



DRC rule file

<pre>layout_primary "ecad"; layout_path "./ecad.gds"; layout_format GDSII;</pre>	;; top layout cell ;; gds file ;; layout format
<pre>results_db -drc drc.ascii;</pre>	;; contains drc output results
	OLY as layer name for layer no 3

```
layer_def CONT 5; ;;defines CONT as layer name for layer no 5
layer_def MET 6; ;;defines MET as layer name for layer no 5
```

```
;; Rule check POLY_CONT_SPACING
```

```
rule "POLY_CONT_SPACING" {
    caption check POLY to contact spacing < 3;
    exte POLY CONT -lt 3 -abut lt 90 -single_point -output region;</pre>
```

ъ

Running PVS from Innovus (DRC)

	Innovus PVS menu
rify	PVS Tools Windows Flows
	Open PVS Run
- @	Recent Runs
	Run DRC
	Run ERC
	Run PERC
	Run LVS
	Run XOR
	Run FastXOR
	Device Signatures
	Run CV
	Run Metal Fill



Running PVS from Innovus: DRC Rules

ile Options "	Tools <u>T</u> oolbars <u>H</u> elp	cādence
🖻 🖬 🗞	🌽 ji 🗟 🗞 🎭 🔁 ji 🛐	
P	Rules	
-	Tech&Rules Configurator Include PVL	-
Run Data	Technology	
0.80	Technology mapping file	Rebad
Rules	Technology no ne	
0	Rule Set	
Input	Ruks	
	Rules File(s)	(Add)
0		(Remove)
Output		Chambre
帮		
DRCOptions		
Start DE		
DRCDE		<u> </u>
PVS RV		
Cancel		
Submit		
Supini		
		No Presets File was Loaded

Running PVS from Innovus: DRC Inputs

e Options Too	ols <u>T</u> oolbars <u>H</u> ep	cādence
🖻 🖬 🗑	🥔 🗟 🇞 🎒 🚱 🛐	
	Layout	
P	Top Cell Name	
Run Data	Create GDSII	(
8×0		
Rules	Merge File(s)	Add
0		Remove
Input		
		View
0		Edit
Output	-Layout Options	
科	Map File streamOut.map	(View)
DRC Options	👱 Uniquify Cell Names 🔛 With Prefix	A CONTRACT OF
	DBU/UU 1000	
	Additional setStreamOutMode Parameters	
🖌 Start DE	Additional streamOut Parameters	
DRCDE		
PVS RV		
	Area to Check on Layout	Select on Viewer
Cancel	Ratten Input Hierarchy	
Apply	👱 Abort on Layout Error	
Su <u>b</u> mit		

Running PVS from Innovus: DRC Outputs

<u>F</u> ile <u>O</u> ptions T	Fools Toolbars Help Cadence
10 🖬 🗑	🚽 🗟 🗞 🕵 😫 😱
	Output
P	Report
Run Data	Name ecad.sum
exe	Limit 1000
Rules	Replace File U Append To File Statistics by Cell
0	Output Format ASCII ecad.drc_errors.ascii
Input	Output Errors Hierarchically Rules Definition
0	Use Waivers for DRC Checks
Output	O DRC DE. S Batch
得	Setup View Edit Load
DRC Options	Waiver File(s)
	View
	Applied Waivers ecad.drc_errors.asci.waivers.applied
	Rejected Walvers ecad drc_errors asci walvers rejected
	Tolerance, UU 0 Magnity_in Save Setup as
	Add Suffix to Cells
	Vertices Per Polygon 4098
🖌 Start DE	Additional Output
DRCDE	Create DFM RV Data
💛 PVS RV	DFM Summary Reports Directory dfmrv_summary
Cancel	_ Criteria File
Apply	Keep Layers None
Submit	

Running PVS from Innovus (LVS)







Running Pegasus from Innovus: LVS Inputs

×.	Create GDSII gds.gz	6
Rukes	Merge File(s)	(A
0		For
Input		
0		
output		<u></u>
鄅		
Options	Layout Options	
	Map File streamOut.map	
	Vith Prefix	
	DBU/UU 1000	
	Additional setStreamOutMode Parameters	
	Additional streamOut Parameters	
	Create SPICEspi	1
	Abort on Layout Error	
	Exclude Comparison Step	
	Schematic	
	🖲 Innovus 🥌 Netlist	Check Sche
	Top Cell Name	
	Create Verilog .v	0
	Netlist File(s) Type Auto	(Ad
		Rem
		<u> </u>
		E
	CPF File	C
	Include Physical Cell Instances	

Running Pegasus from Innovus: LVS outputs

<u>File</u> Options Too	is <u>T</u> oolbars <u>H</u> eip	cādence
11 🕞 🔂	🔎 🖻 🗞 🍢 🗐 🛐	
	Output	
e e e e e e e e e e e e e e e e e e e	H-Cell Settings	
Run Data	Automatch	
**	HCell	
Rules	GenHierCells	
0	Z Run ERC Checks	
	ERC Report	
Input	Name PLL_250MHZ.sum	
0	Limit 1000	
Output	🖲 Replace File 🥥 Append To File 🔛 Statistics by Cell	
科	Output Format ASCII PLL_250MHZ.erc_errors.ascii	
LVS Options	Output Errors Hierarchically Rules Definition 🔽 🔄 Output Complete Pathchk Results	
	Use Waivers for ERC Checks	
	CLVS Report	
	Name PLL_250MHZ.rep	
	Options -none	SET
Start LVS DE 👱	Limit 50 Mismatched Nets Limit 100	
(<u>C</u> ancel	Additional Output	
	🔜 Create Quantus QRC Input Data	
Submit	QRCDataDir svdb	
	No Presets	File was Loaded

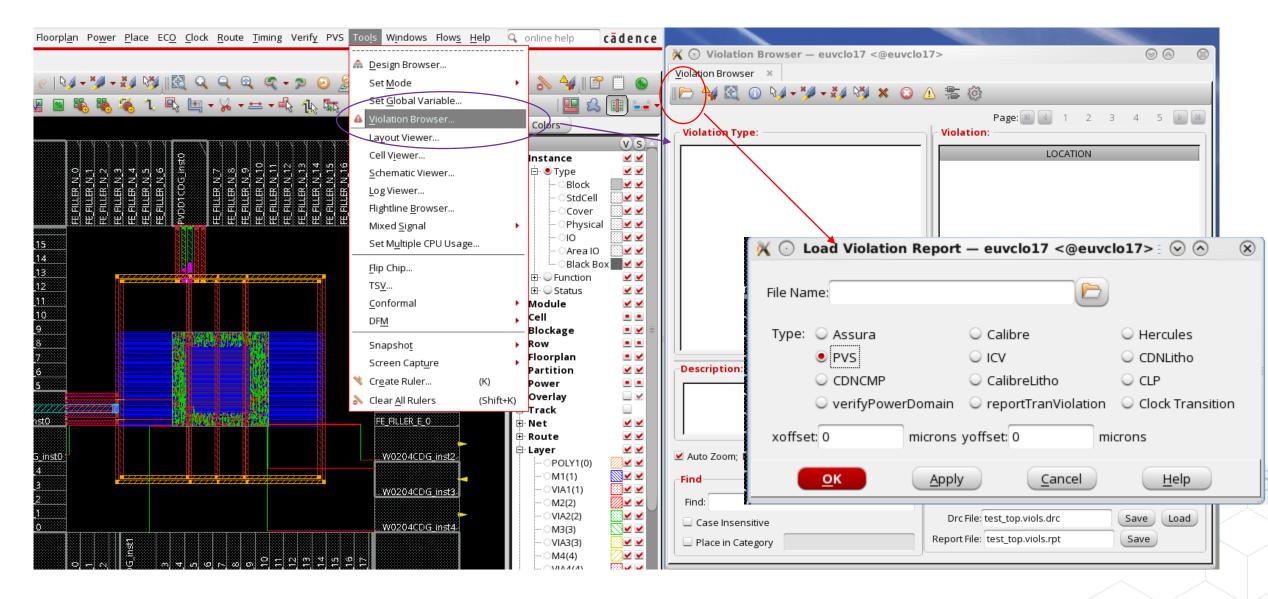
Main report window

This section shows log of the current job. Errors and warnings are automatically highlighted in this field:

```
''''' Start processing file 'lvs.rul.__pre__' '''''''
LVS_REPORT_FILE "lvsrep";
[WARN]: LVS_REPORT_FILE is skipped, it is set in control file.
LVS_REPORT_MAX 1000;
[WARN]: LVS_REPORT_MAX is skipped, it is set in control file.
MASK_RESULTS_DB -none;
MASK_SVDB_DIR svdb -query -cci;
FLOW_DATA ui_data;
LVS_FIND_SHORTS yes;
```

finished committing cdl netlists (0s)
ERROR (NVN-15200): schematic_primary "ecad" - cell 'ecad' does not exist in the schematic side.
Check the rule file and rerun it.
Generating the LVS report...
Command "pvsnvn lvs.rul.rsf" was terminated with signal ll
[ERROR] pvsnvn terminated abnormally.

Load results in Innovus



Loads results in Innovus

ѷӳ + ЎЎ + ЎЎ 🕅 🔍 🔍 🔍 🔍 🗢 🦻 🥹 + 🕭 🎲 💷 🏯 🕼 🛈		
🖻 🗞 🧞 1. 🖳 🖽 - 🖌 - 🖴 - 🔩 🏤 🛼	I 🔛 幺 💷 🖬 🖬	
	All Colors 🔀 📀 Violation Browser — euvclo17 <@euvclo17>	\odot \odot \otimes
	Violation Browser ×	
	Image: Image	¢۶
	⊕ Module	Page: 🛛 🚽 1 2 3 4 5 🕞 🕅
	E-Cell Violation Type:	Violation:
	E. Cell	LAYER OBJECT1 OBJECT2 L
	E- Geometry (4/4) E- Blockay E- Short (3/3)	M8(8) INST PDDW0204CDG_inst4 INST FE_FILLER_S_25 (55
	⊡ Overlay	
	⊕-Track ⊕-Net	
	🗄 Route	
×	De Layer	
	OM1	
	- OVIA Description:	
	M3 VIA Pin of Cell PDDW0204CDG_inst4 & Blockage of Cell FE_FILLER_S_	25
	OM4 Actual: 1.0965 Min: 2	
PDBW0204CDG insta		
		Save Report
		Drc File: test_top.viols.drc Save Load
	Bump Case Insensitive	Report File: test_top.viols.rpt Save
	Detail U Place in Category	



Simulation



Prerequisites for final simulation

- The final schematic has been laid out and has passed DRC and LVS checks and the parasitics for the block have been extracted.
- The RC extraction contains not only the designed devices and their connectivity but also a network of parasitic devices (Rs, Cs, Ls) that model the physical routing between the devices.
- Since the parasitic simulations can be time-consuming, these analyses are limited here to specific and necessary parameters, testbenches, and process corners.

Cadence Mixed-Signal Solution : AMS Designer

- Combine the power of the Spectre[®] and Xcelium[™] simulators, now with AMS-Flex for easier version control
- Supports extensive set of languages that can be mixed and matched according to need
 - Spice, Verilog + A/MS, VHDL +AMS, SystemVerilog, Verilog AMS & SV RNM, C++ representations
- Common AMS simulator within *domain specific* environments
 - Use with Virtuoso[®] ADE Assembler for simulation management and distribution with an analog point of view
 - Use with Cadence SimVision[™] Debug tools for full range of digital analysis
- Low power simulation with IEEE 1801

Easy Use Model (Virtuoso GUI) Starting from Scratch in 5 steps

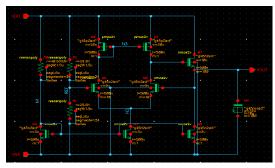
Schematic (Creation	> Maestro	Creation

Partitioning Configuration

ADE Setup

Interactive Debugging

• Virtuoso Schematic Editor



- Language text Importing into Virtuoso
- ADE Advanced TestBench (valuable for digital super top or UVM-MS)

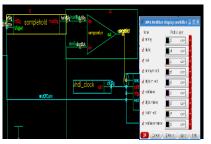
	And Configuration have	Cathorn		
urbar	ich Top Cell Neme	tent_tox		
rign	top Configuration Made	्र त्व 🔹	Instance	
rign	Ing Cell Name	yapply		
and the second	Lame for Testbench	Inst_Los	- 1855	
erritga	ealter Mormation for Parts In Para	fict to the Design	Top	
Enabl	collanzance (terms	7/15	Rule Type	Configuration Rale
1	F2V_ams	CE1	10121	omaib
~	tert Jang BHZ	instance.	1.10	PLUDOP CIK, NO
*	bbdk pill sco	<01	1.9C	PLL LOOP block pill yes
	14T_J10031851	intance.	1010	PULL009
1	CPUMP	a et 1	Infaited.	PT100P
-	L25	cp1	V26	PLL LOOP LPT
	ernons princips for TicsRecrub			
			Irun Optiana	
Enabl				
Erati *	 4 Acompibilis Click page to add new your and 			

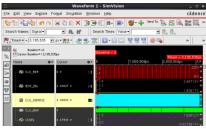
unch <u>F</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp			cuu	e n c
🗅 🗁 🔒 🖓 🚸 🤌 🦿 🔘 🗁 🖒 🖧	🙆 🔍 s	arch 🔽	- 0	
op Cell	? 5 ×	Global Bindin	ngs	? 8
ibrary: amsPLL		Library List:	amsPLL linterLib	(
				_
Cell: PLL_160MHZ_sim		View List:	loga vhdi vhdiams symbol	
/iew: schematic				
		Stop List:	spectre spice symbol	
Open Edit ADE L ADE Exp	plorer	Constraint List:		
	_			-
		View To Use	aberited View I	u A
Target: Instance				
Target: Instance	_	View To Use	herited View L	Lie 🛆
		View To Use	herited View L	
Instance D- A (amsPLL PLL_160MHZ_sim schematic)		View To Use	herited View L	
Instance (amsPLL PLL_160MHZ_sim schematic) G (analogLib tccs spectre) (analogLib tccs spectre) (analogLib tcs spectre)		View To Use		
Instance Instance Profile Construction P	_	View To Use	spectre spice .	
Instance Ins		View To Use	spectre spice . spectre spice .	
Instance → ① Figurals2LPL_Instance → ① Figurals2Lb (ccs spectre) → ② Figurals2Lb (ccs spectre) → ③ 0 analogLb id (spectre) → ③ 0 (analogLb id (spectre) → ③ 0 (angle)2D (spectre) → ③ 0 (angle)2D (spectre) → ③ 0 (angle)2D (spectre) → ④ 0 (angle)2D (spectre) → ④ 0 (angle)2D (spectre) → ④ 0 (angle)2D (spectre) → ⑤ 0 (angle)2D (spectre) → ⑥ 0 (angle)2D (spectre) → ⑥ 0 (angle)2D (spectre) → ◎ 0 (angle)2D (spectre		View To Use	spectre spice . spectre spice . spectre spice .	
Instance ■ ① [amsPLL PL_10ML2_smschematic) ■ ② [analogLb cccs spectre] ■ ② [analogLb ccs spectre] ■ ② [amsPLL PL_10MML2 schematic) = ③ [angd009 mossach spectre) = □ 11 (amsPLL PL_FPD schematic) = □ 11 (amsPLL PL_FPD schematic)	-	View To Use	spectre spice . spectre spice . spectre spice . spectre spice .	
Instance Image: Image: Instance Image: Image: Instance Image: Image: Instance Image: Imag		View To Use	spectre spice . spectre spice . spectre spice . spectre spice . spectre spice .	
Instance Image: Inst		View To Use	spectre spice . spectre spice . spectre spice . spectre spice . spectre spice . spectre spice .	
Instance Ins		View To Use	spectre spice . spectre spice . spectre spice . spectre spice . spectre spice . spectre spice .	
Instance Image: Inst		View To Use	spectre spice spectre spice spectre spice spectre spice spectre spice spectre spice spectre spice	
Instance Ins			spectre spice. spectre spice. spectre spice. spectre spice. spectre spice. spectre spice. spectre spice. spectre spice.	
Instance ■ mmPLL PLL MEMTL2, sm schematic) ■ Fit (an 160 MHZ, sm schematic) ■ Fit (an schematic) ■ Fit (anst)L PLL PO schematic) ■ Fit (anst)L PLL PO schematic) B Fit (anst)L PLL NOT schematic)	functional		spectre spice. spectre spice. spectre spice. spectre spice. spectre spice. spectre spice. spectre spice. spectre spice. spectre spice. spectre spice.	
Instance ■ A (amsPLL PL_IOMEX_sms schematic). ■ F (analogUb cccs spectre) ■ D (analogUb dccs spectre) ■ D (anselUp L) PL (DMMS schematic) ■ D (anselUp L) PL (DMS schematic) ■ D (anselUp L) PL (DMS schematic) ■ D (anselUp L) PL (DMMZ (L) schematic) ■ D (anselUp L) PL (DMMZ (L) schematic) ■ D (anselUp L) PL (DMMZ (L) schematic) ■ D (anselUp L) PL (AST schematic) ■ D (anselUp L) PL (AST schematic) ■ D (anselUp L) PL (DMMZ (DMZ vorter) ■ D (anselUp L) PL (DMZ (DMZ vorter)	functional		spectre spice spectre spice	
Amarghi Pil, 1904HZ, sm schematic) Originalization (Comparison (Comparison)) Originalization (Comparison) Originali (Comparison) Originalization (Comparison) Originalizatio	functional		spectre spice spectre spice	
Instance ■ Rimal@LthUt_sim schematic). ■ Figural@LthUt_sim schematic). ■ Figural@LthUt_simstrict schematic) ■ Bigard@LthUt_schematic) ■ Bigard@Lthut	functional		spectre spice gpectre spice	
Instance Image: ILL_IND/MR_sem schematic) Image: ILL_IND/MR_sem schematic) Image: ILL_IND/MR_sem schematic) Image: ILL_IND/MR_schematic) Image: ILL_IND/MR_schematic) Image: ILL_IND/MR_schematic) Image: Image: ILL_IND/MR_schematic) Image: Image: ILL_IND/MR_schematic) Image: Imag	functional		spectre spice spectre spice	

Namespace: CDBA Filter







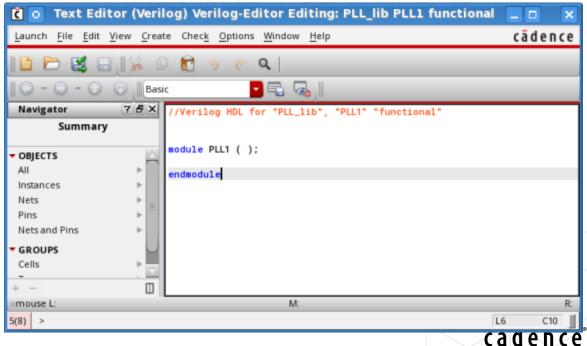


cadence

Importing Verilog into Virtuoso Using the New Cellview Form

- 1. Create a new cellview in the Library Manager for the chosen design library.
 - You can set the *View* name to any name.
 - Choose Verilog as the *Type*.
 - Choose Text Editor as the Application to Open with.
- 2. When you click OK on this form, a Verilog text editor comes up. Copy-paste your Verilog text file here.
- 3. Once edits are done, <u>save</u> and <u>quit</u> this text file, which triggers compilation.
 - If a symbol doesn't exist, you will be prompted to create one. If it does exist, then port consistency is verified.
- Once compiled, the file is copied into the DFII library structure and visible in the Library Manager as a functional (default name) view. You are also be notified of any errors in the CIW.





Importing Verilog into Virtuoso Using the Verilog-In Form

- Go to the Command Interpreter Window (CIW) in Virtuoso and choose File – Import – Verilog to open the Verilog In form.
- 2. <u>Browse</u> and add your Verilog file to the Verilog Files to Import section.
- 3. Set the Target Library Name, and add Reference Libraries if needed.
- 4. Specify the Functional View Name for the Functional blocks.
- Optionally, in the Global Net Options tab on the top of this form, enter the global net names for power/gnd used during schematic creation.
- 6. Click OK to trigger compilation, you can review a log file if you choose to.
- 7. View the new cell view that is created in the Library Manager.

C and C 2022 Cade®ce Design Systems, Inc. All rights reserved

	/erilog In ? _		X
mport Options Global Net Opt	ons Schematic Generation Options		
Verilog Files To Import	ISDesigner_19_09/ADE1ab/diglib/PD.v)	
Target Library Name			
Reference Libraries			
Reference Symbol View Names			
	5,		
Overwrite Options			
 Import Modules as 			
Structural Modules	schematic 🔽		
Verilog Cell Modules	Create Symbol Only		
Schematic View Name	schematic		
Netlist View Name	netlist		
Functional View Name	functional		≣
Symbol View Name	symbol		
Filter Modules			
Library Pre-Compilation Opt	ions		
• Other Input Options			
- Other Output Options			
Log File	./verilogIn.log		
Work Area	/tmp		
	./verilogIn.map.table		
Name Map Table			

Referencing Text Files Outside DFII: Using –v/–y

- From the ADE, go to Simulation Options AMS Simulator to open the AMS Options form.
- 2. Click the Include Option Settings button to open the Include Options Setup form.
- 3. In this form, you can browse to and add the required files or library.

🗙 💿 AMS Options 🛛 🔹 📮 🗖 🗙									×	
Main	Netlister	Messages	PLI	SDF	Timing	Low Power	UVM	Miscellane	ous	
INCLUDE OPTIONS									Â	
Include Option Settings										
TIM	ESCALE OPT	IONS							=	
Global sim time					1					
Units for global sim time						ns				
Global sim precision						1				
Units for global sim precision					ns					
DISCIPLINE OPTIONS										
Default discipline					logic					
Use detailed discipline resolution										
1				<u>о</u> к	<u>C</u> ancel	<u>D</u> efaults		Apply H	elp	

🗙 o ams0: Include Options Setup	? _ 🗆	×
Libraries Files/Directories Include Options		
On/Off Library Files/Directories		
白 Library Files (-v)		
Click here to add library files>		
🗄 Library Directories (-y)		
	_	
1		
Lib Extension (for -y only)		
<u>OK</u> <u>C</u> ancel		lp.

Referencing Text Files Outside DFII: Using Library Compilation Options (–makelib/-reflib)

Use the **—makelib** option to precompile design units in the specified files into a reference library.

Using the -makelib Option:

- You can pre-compile "golden" digital design units into a reference library using the -makelib option.
 - When top-level design files are compiled, the reference library is scanned for components instantiated in the design.
- This pre-compiled library is reusable and you can include the compiled cells in your mixed-signal simulations.

X o	AMS Op	otions						? 📜		×
Main	Netlister	Messages	PLI	SDF	Timing	Low Power	UVM	Misc	ellane	ous
INCL	UDE OPTIO	NS								_
			•							-
			Inc	lude O	ption Sett	ings				
Т	o ams0:	Runtime Li	brarv	Comp	ilation S	ettings (-ma	kelib)	2		×
GI		ry Path[:Name]		celib sour		_	-makelib (option		
E	-makelib		<cli< td=""><td>ck here t</td><td>o add -make</td><td>lib files></td><td></td><td></td><td></td><td></td></cli<>	ck here t	o add -make	lib files>				
Un	1	List sourc	e fil	e na	mes he	ere along	with		J.	=
Gl						ne in whic	h it		×	
Un	1	has to be	cor	npile	d					
-										
	enable -mixes	SC .								
De	enable -smart	order								
Us										
						<u>о</u> к <u>с</u> а	ncel	<u>A</u> pply	<u>H</u> el	
LIBR	ARY COMPI	LATION OPTIC	DNS							
		Pre-o	:ompi	led libr	aries (-ref	lib) Settings				
		Due tiere			ilation (m					
		Run-time	norar	y comp	mation (-n	nakelib) Settin	gs			27
				ок	Cancel	Defaults		Apply	Сн	elp

Referencing Text Files Outside DFII: Using Library Compilation Options (–makelib/-reflib)

Using the **-reflib** option, you can specify the library directory containing precompiled source files that can be referenced during simulation.

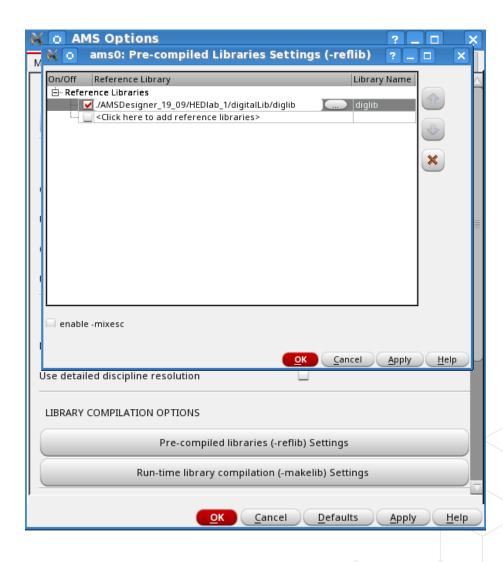
Using –reflib Option:

e Design Systems, Inc. All rights reserved

cadence

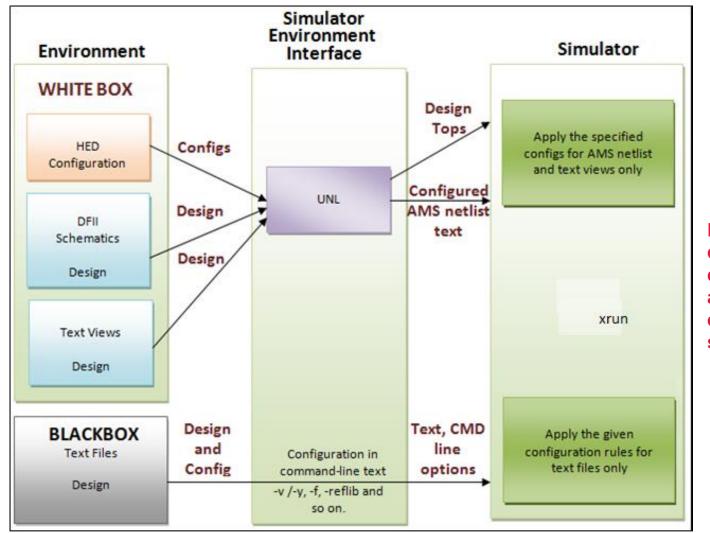
- Clicking the Pre-compiled libraries (-reflib) Settings button opens a form with a similar name.
- Browse and specify your precompiled library in the provided field.

<u>Important:</u> When using -reflib, we assume that the text libraries were pre-compiled external to Virtuoso using the *xrun* -makelib command.



WDUs and BDUs: Example

The ADE AMS UNL flow comprises a netlisting phase using the Unified Netlister, followed by the binding phase during elaboration and then finally simulation using the xrun executable.



BDUs and WDUs are considered together during elaboration and the complete design config is simulated.

© Cadence Design Systems, Inc. All rights reserved d e n c e35



Day 3 Conclusion





What we did cover

- Chip assemble, AoT, DoT
- STA analysis for mixed signal design

- Power analysis
 - Power
 - Resistance
 - IRDROP
 - EM
- Physical checks
 - DRC
 - LVS
- Simulation

To go deeper on part 3, useful labs

Static Timing Analysis (Signoff Timing Analysis) using Tempus 20.1 (08 Oct 2020)

Static Timing Analysis on Schematic-based Mixed-Signal Design (Common UI) (29 Jul 2022)

Power and Rail Analysis Using Voltus 20.13 (Signoff Power Analysis) (15 Mar 2021)

Running Pegasus DRC, LVS within Innovus Implementation System (21 May 2021)

Pegasus Hierarchical Metal Fill Flow in Innovus: Overview (05 Mar 2021)

ECO on Schematic-based Mixed-Signal Design - Stylus Version (29 Aug 2022)

cādence

© 2022 Cadence Design Systems, Inc. All rights reserved worldwide. Cadence, the Cadence logo, and the other Cadence marks found at https://www.cadence.com/go/trademarks are trademarks or registered trademarks of Cadence Design Systems, Inc. Accellera and SystemC are trademarks of Accellera Systems Initiative Inc. All Arm products are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All MIPI specifications are registered trademarks or trademarks or service marks owned by MIPI Alliance. All PCI-SIG specifications are registered trademarks or trademarks are the property of their respective owners.