StarRC Q-2019.12

Beta Update Training

StarRC BU

Oct 14, 2019

CONFIDENTIAL INFORMATION

The following material is confidential information of Synopsys and is being disclosed to you pursuant to a non-disclosure agreement between you or your employer and Synopsys. The material being disclosed may only be used as permitted under such non-disclosure agreement.

IMPORTANT NOTICE

In the event information in this presentation reflects Synopsys' future plans, such plans are as of the date of this presentation and are subject to change. Synopsys is not obligated to develop the products with the features and functionality discussed in these materials. In any event, Synopsys' products may be offered and purchased only pursuant to an authorized quote and purchase order or a mutually agreed upon written contract.

Q-2019.12 Update Training

- **New Features**
  - Gate-Level Flows
  - Tx-level Flows
  - Other Features Common To Both Flows
- **Appendix**

SYNOPSYS®

New Features in Q-2019.12 for Gate-Level Flow

- Gate-Level Performance Improvement
- Virtual Metal Fill Support
- Instance Port Location Closer To The Driver
- Parasitic Explorer –Selected Nets Short Visualization
- Parasitic Explorer – GUI
- LEF/DEF Based 3DIC WoW

Q-2019.12 Gate-Level performance Improvement

Targeted on designs with SMC extraction and GPD output

- 1.3x speedup for SMC vs 2019.03
  - Blocks (<=7nm), LEF/DEF or NDM
  - Runtime Improvement
    - **Pre-Extract**
      - SMC setup stage optimization
      - PreXT DP scheduler optimization
      - Support NDM patterned metals for PG shape
    - **Extraction**
      - Layer-based rigorous etch
      - Resistance extraction performance <=5nm
      - Thickness variation modeling and caching optimization
  - Memory Improvement
    - Memory reduction on large design <=7nm
    - SMC: 30%
    - SC: 13%

SMC runtime Improvement

SMC	prext	extract	postxt	overall
Ratio	30%	65%	5%	100%
All	1.16	1.09	0.91	1.11
<=7	1.62	1.16	0.93	1.28
<=16	1.28	1.10	0.92	1.16
>16	0.99	1.05	0.90	1.03
Blocks	1.23	1.10	0.92	1.14
Hier	1.03	1.06	0.88	1.04
Flat	1.05	1.06	0.95	1.05
LEF-DEF	1.11	1.08	0.91	1.09
NDM	1.22	1.09	0.92	1.14
LEF-DEF <=7	**1.47**	**1.18**	**0.94**	**1.24**
NDM <=7	**1.72**	**1.14**	**0.92**	**1.31**

SC(Single Corner) runtime Improvement

SC	prext	extract	postxt	overall
Ratio	32%	64%	4%	100%
All	1.00	1.08	0.88	1.05
<=7	1.06	1.14	0.91	1.11
<=16	1.03	1.10	0.89	1.07
>16	0.95	1.03	0.86	1.00
Blocks	1.01	1.08	0.88	1.05
Hier	0.96	1.04	0.87	1.01
Flat	1.10	1.06	0.93	1.07
LEF-DEF	1.01	1.08	0.88	1.05
NDM	1.00	1.08	0.88	1.04
LEF-DEF <=7	**1.05**	**1.16**	**0.92**	**1.13**
NDM <=7	**1.06**	**1.13**	**0.90**	**1.10**

No new or changed commands needed

New Features in Q-2019.12 for Gate-Level Flow

- Gate-Level Performance Improvement
- Virtual Metal Fill Support
- Instance Port Location Closer To The Driver
- Parasitic Explorer –Selected Nets Short Visualization
- Parasitic Explorer – GUI
- LEF/DEF Based 3DIC WoW

Virtual Metal Fill Support

For Gate level only. No virtual metal fill with real metal fill in one extraction is allowed

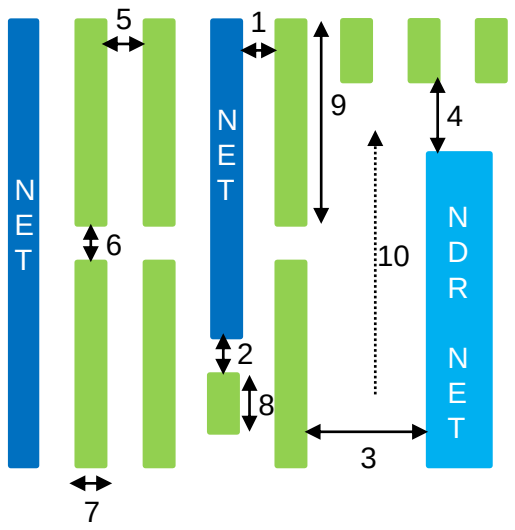
- Best-in-class correlation with real metal fill with small performance footprint
- No need to update ITF/NXTGRD
- Available in StarRC stand-alone and StarRC In-Design (in IC Compiler II).
- Fully parameterized virtual track fill generation, including non-default-rule (NDR) nets specialization

StarRC Command Options

VIRTUAL_METAL_FILL_POLYGON_HANDLING:
[IGNORE] | FLOATING | GROUNDED

VIRTUAL_METAL_FILL_PARAMETER_FILE: <file path>

VIRTUAL_METAL_FILL_NDR_NETS: <net1> <net2>…



	Parameter Name
0	db_layer_name
1	net_fill_w_spacing
2	net_fill_l_spacing
3	ndr_net_fill_w_spacing
4	ndr_net_fill_l_spacing
5	fill_fill_w_spacing
6	fill_fill_l_spacing
7	fill_width
8	min_fill_length
9	max_fill_length
10	direction (V|H)

SYNOPSYS®

VIRTUAL_METAL_FILL_PARAMETER_FILE

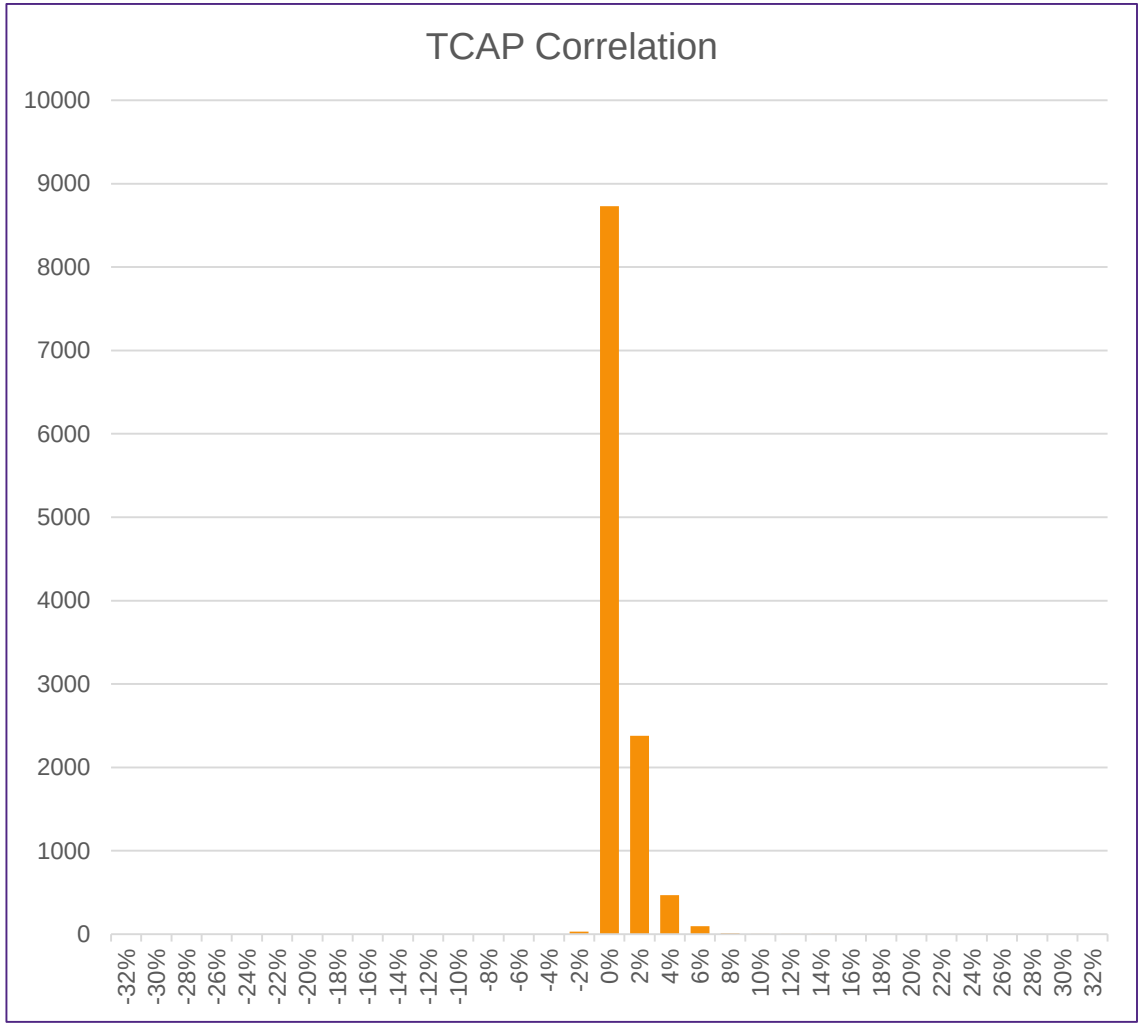
Parameter name	Type	Accepted values	Description
db_layer_name	string	Any valid db layer name	Name of the dbLayer on which user wants to generate virtual metal fill
direction	char	(V,H)	Routing direction of virtual metal fill shapes (vertical V, horizontal H)
fill_width	Float (nm)	[WMIN, inf)	Width (fixed) of virtual metal fill shapes in the direction perpendicular to the routing direction
min_fill_length	Float (nm)	[WMIN, inf)	Min length of virtual metal fill shapes in the direction parallel to the routing direction
max_fill_length	Float (nm)	[min_fill_length, inf)	Max allowed length of virtual metal fill shapes in the direction parallel to the routing direction
min_fill_route_w_spacing	Float (nm)	[SMIN, inf)	Min spacing between virtual metal fill and design shapes in the direction perpendicular to the routing direction
fill_route_l_spacing	Float (nm)	[SMIN, inf)	Spacing between virtual metal fill and design shapes in the direction parallel to the routing direction
min_fill_fill_w_spacing	Float (nm)	[SMIN, inf)	Min spacing between virtual metal fill and virtual metal fill shapes in the direction perpendicular to the routing direction
fill_fill_l_spacing	Float (nm)	[SMIN, inf)	Spacing between virtual metal fill and virtual metal fill shapes in the direction parallel to the routing direction
min_fill_blockage_w_spacing	Float (nm)	[SMIN, inf)	Min spacing between virtual metal fill and design blockage shapes in the direction perpendicular to the routing direction
fill_blockage_l_spacing	Float (nm)	[SMIN, inf)	Spacing between virtual metal fill and design blockage shapes in the direction parallel to the routing direction

VMF and Real MF Correlation Result

Good RC correlation b/t VMF and Real MF

- Test case: FinFET process node

```
-------------------- RC Correlation Overview --------------------
Total cap     (C) mean error (abs = 3.000fF): 0.656%
                  std error (abs = 3.000fF): 1.148%
Coupling cap (CC) mean error (abs = 0.300fF, rel = 0.100): -
0.191%
                  std error (abs = 0.300fF, rel = 0.100): 0.303%
Pin-Pin res (P2P) mean error (abs = 50.000Ohm): -0.008%
                  std error (abs = 50.000Ohm): 0.037%
-------------------- TCAP Distribution --------------------
TCAP threshold: 3.000000 fF
Min Error: -2.388%      Max Error: 14.407%
Mean Error: 0.656%      Standard dev: 1.148%
Number of matched nets: 11716
```



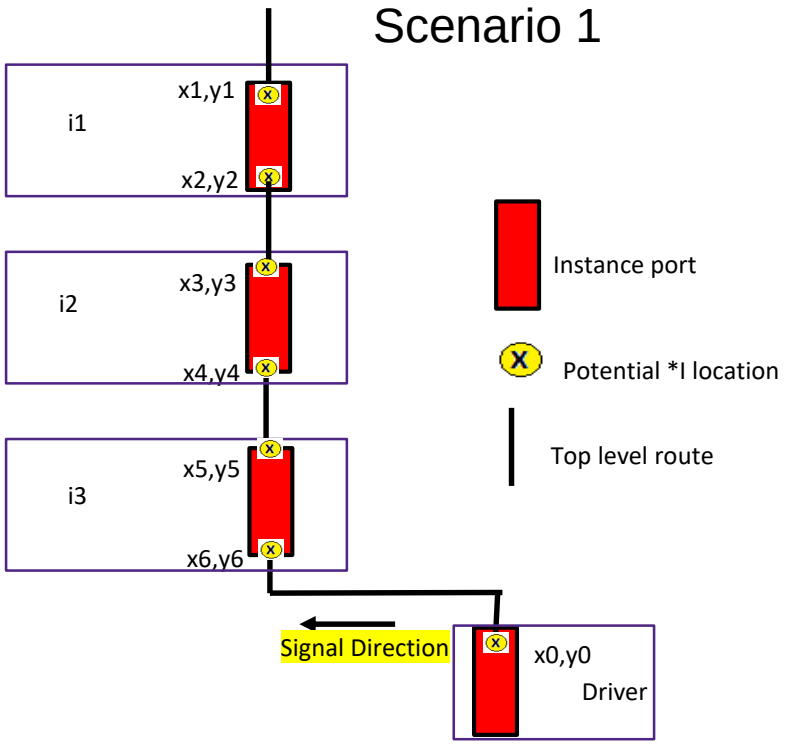
SYNOPSYS®

New Features in Q-2019.12 for Gate-Level Flow

- Gate-Level Performance Improvement
- Virtual Metal Fill Support
- Instance Port Location Closer To The Driver
- Parasitic Explorer –Selected Nets Short Visualization
- Parasitic Explorer – GUI
- LEF/DEF Based 3DIC WoW

Instance Port location closer to the driver

- Existing approach of selecting multiple potential instance port location uses geometrical rules: the priority is given to the highest layer, then lowest coordinate, then left-most coordinate.
- The new approach will select the instance port location closer to the driver, hence the smallest point to point resistance from the driver to all potential location of the receiver instance ports



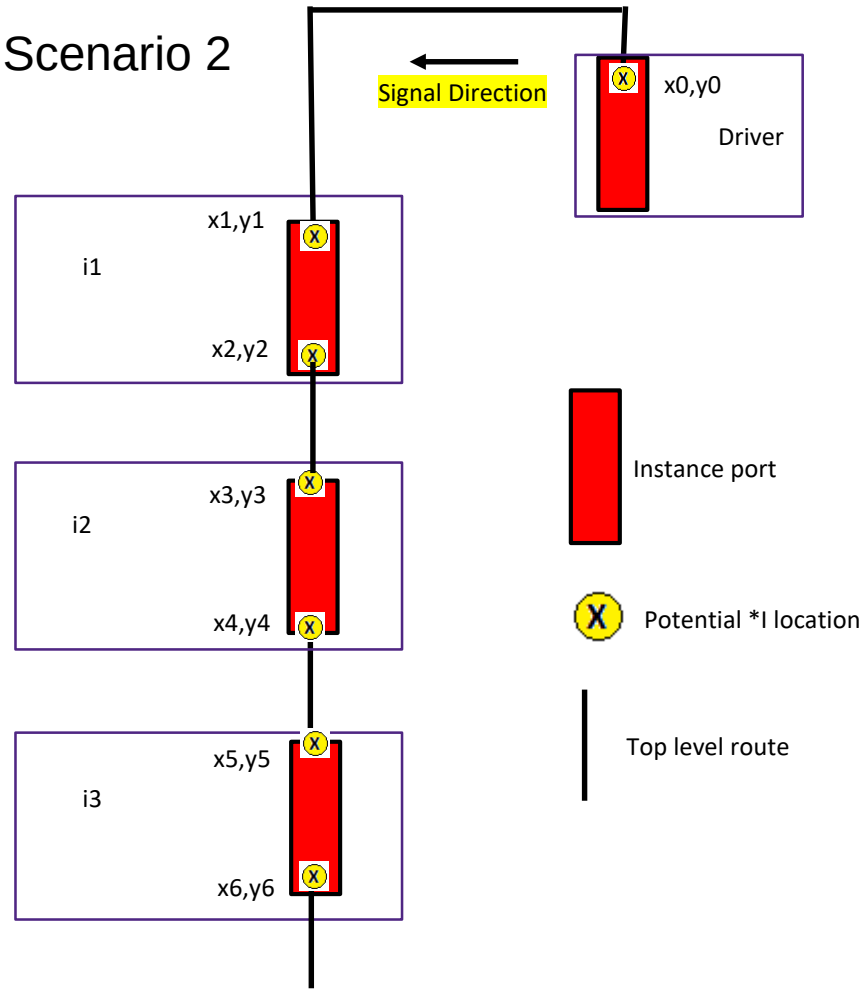
X are potential *I locations, based on interaction point of top level route and instance port of the skip_cell.

Based on the consistent instance port location, (x2,y2), (x4,y4) and (x6,y6) are selected.

In this case since the instance port is a feedthrough net/used as routing resource, the resistance of the instance port is extracted with INSTANCE_PORT:CONDUCTIVE, but does not get added to p2p path from driver (x0,y0) to the receiver *I.

SYNOPSYS®

Instance Port location closer to the driver



In this case, based on the existing approach, (x2,y2), (x4,y4) and (x6,y6) are selected for receiver instance port locations

The resistance of the instance port gets added to p2p path from driver (x0,y0) to the receiver *I (double counted).

In such scenarios, we need to place *I location before the port resistance / closest to the driver, to avoid double counting. So (x1,y1), (x3,y3) and (x5,y5) need to be selected.

StarRC is enhanced to take this into account with a new command

`INSTANCE_PORT_LOCATION_CLOSER_TO_DRIVER: YES`
(default: NO)

With this feature turned on, we can expect to see change in p2p resistance when comparing to previous version, as some of the *I locations might have changed.

This feature is currently supported only in gate level flows.

SYNOPSYS®

New Features in Q-2019.12 for Gate-Level Flow

- Gate-Level Performance Improvement
- Virtual Metal Fill Support
- Instance Port Location Closer To The Driver
- Parasitic Explorer –Selected Nets Short Visualization
- Parasitic Explorer – GUI
- LEF/DEF Based 3DIC WoW

Selected Net Short Visualization

Basic Flow

- To view shorts in a region for a select set of nets, "-write_short_regions" is developed

```
% StarXtract -write_short_regions -nets_file nets_file star_cmd
```

- To create error file, use starrc_shell command as follows

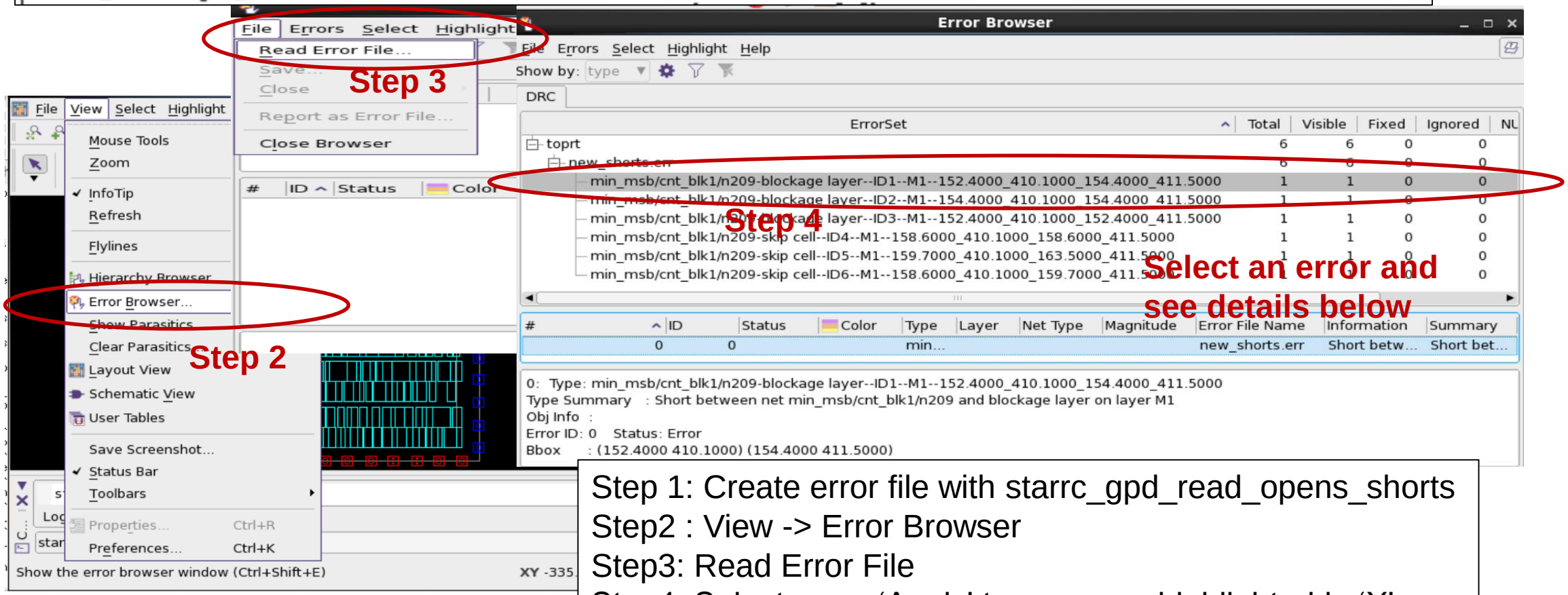
```
% starrc_shell> starrc_gpd_read_opens_shorts –gpd <gpd> -nets "net1 ... net_n"
                -error_file new_short.err
```


Short Region Visualization

Error view from "-write_short_regions" for 1 net

```
starrc_shell> starrc_gpd_read_opens_shorts -gpd toprt.gpd/ -nets min_msb/cnt_blk1/n209 -error_file new_shorts.err
INFO: new_shorts.err exists, removing previous version of the file
INFO: Creating error file: new_shorts.err
INFO: Parsing opens summary file
INFO: Parsing shorts summary file
starrc_shell> gui_start
```

Step 1

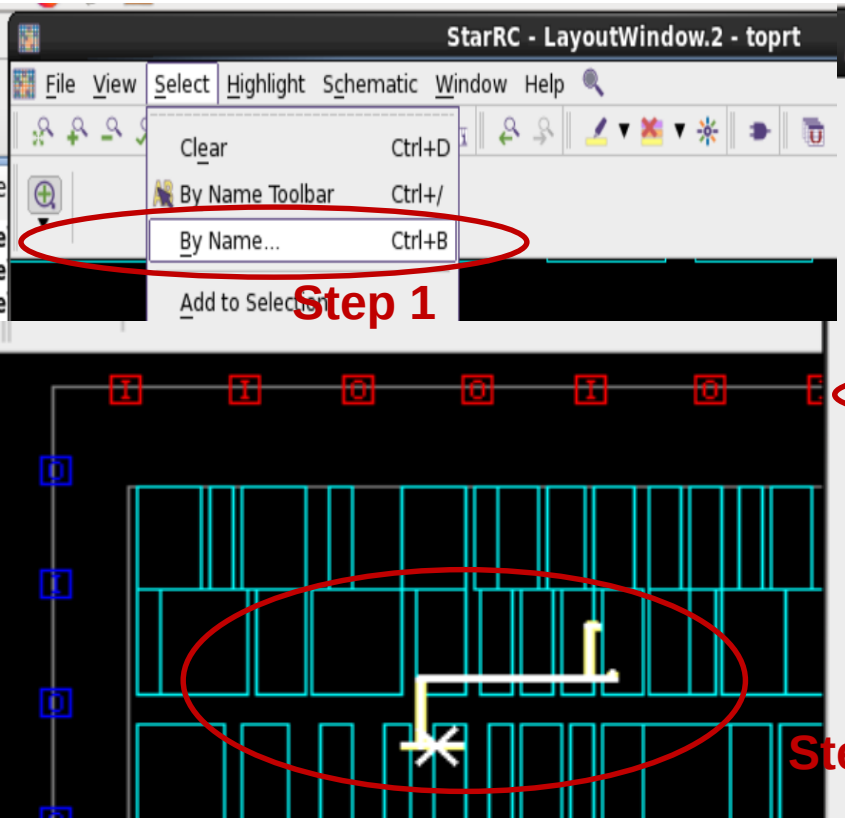


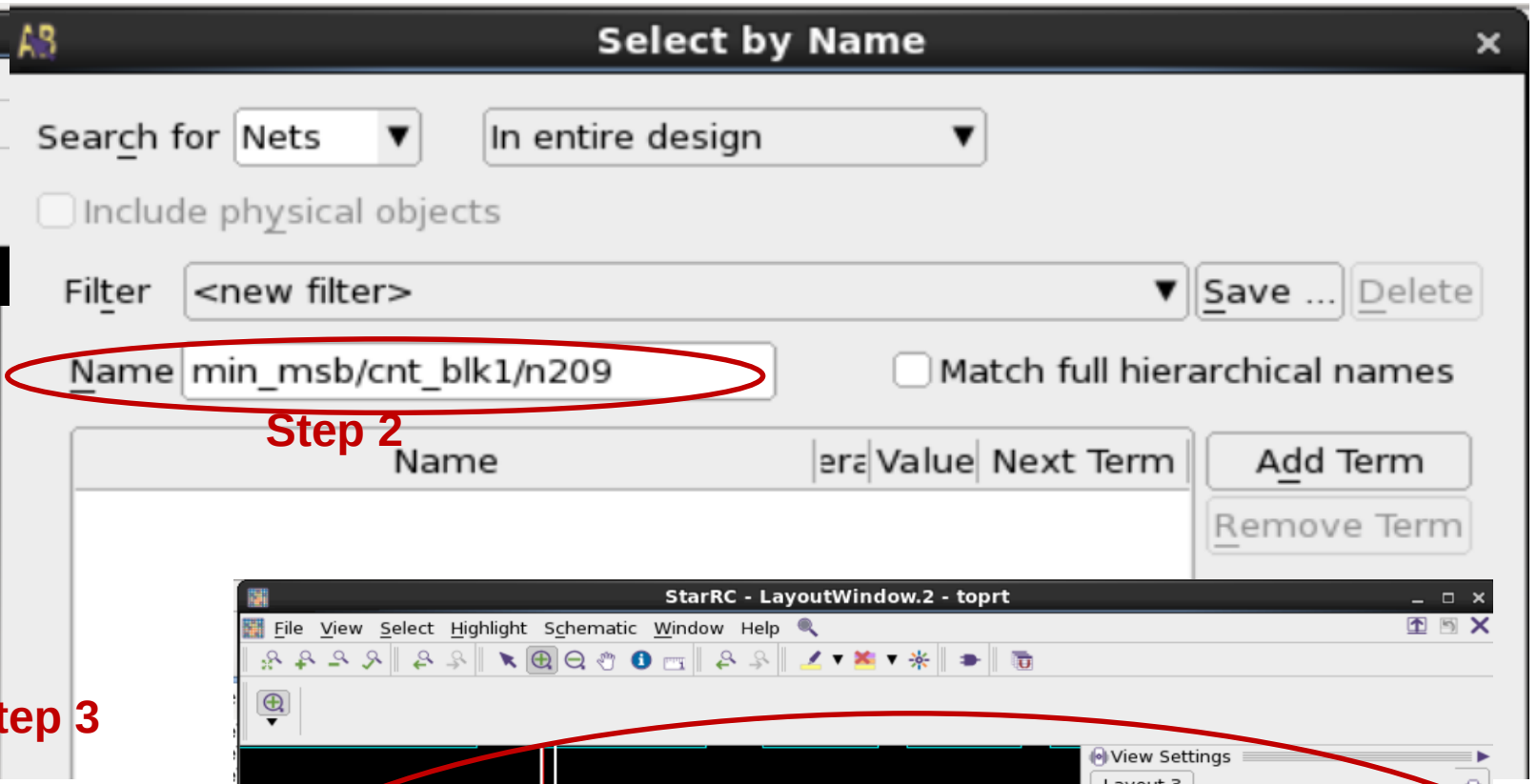
Step 1: Create error file with starrc_gpd_read_opens_shorts
Step2 : View -> Error Browser
Step3: Read Error File
Step4: Select error, 'Apply' to see error highlighted in 'X'

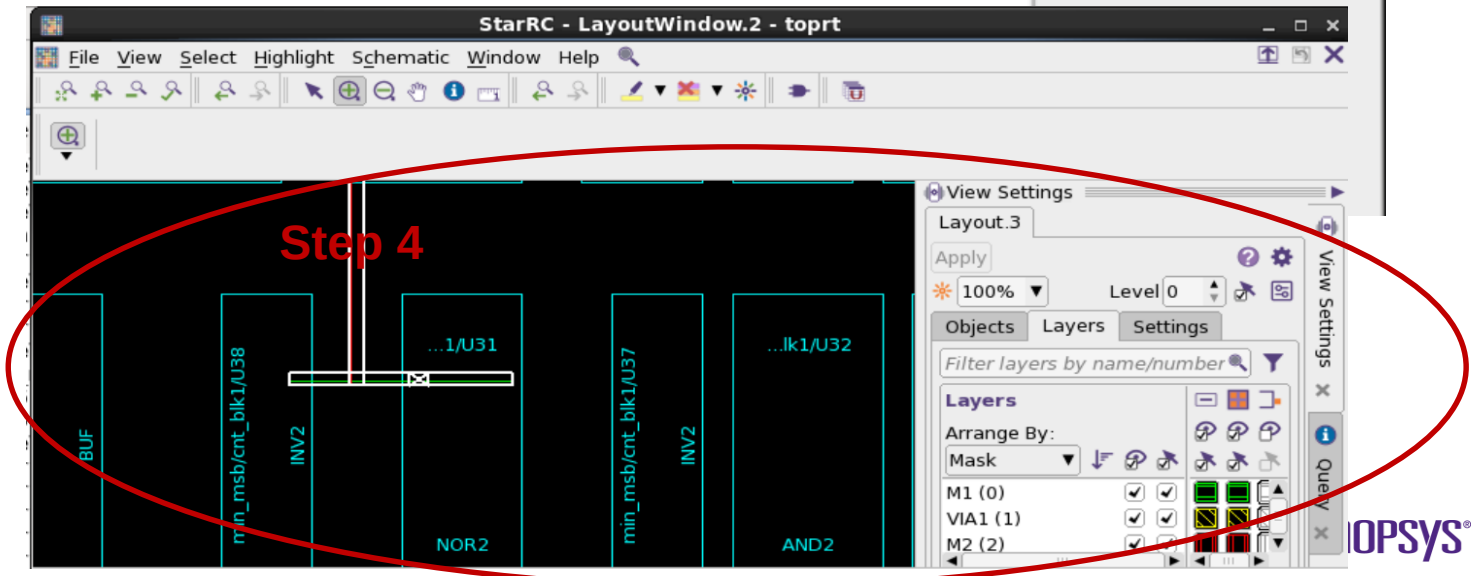
SYNOPSYS®

Short Region Visualization

View error and highlight net at the same time







Step 1: Select -> By Name
Step2 : Provide net name for “Nets”
Step3: Highlighted net and error in X
Step4: Zoom and see layers

New Features in Q-2019.12 for Gate-Level Flow

- Gate-Level Performance Improvement
- Virtual Metal Fill Support
- Instance Port Location Closer To The Driver
- Parasitic Explorer –Selected Nets Short Visualization
- Parasitic Explorer – GUI
- LEF/DEF Based 3DIC WoW

Parasitic Explorer GUI

Introduction

- Allows display of Parasitic Map and layout (LEF/DEF & NDM)
- Supports layer based query for selected nets
- Supports very fast loading of Errors

Parasitic Explorer Flow and Usage in starrc_shell

GUI Invoke

1. Run extraction with the following command

   PARASITIC_EXPLORER_ENABLE_ANALYSIS: YES

2. Invoke the StarRC shell

   % starrc_shell

3. For SMC designs, set the parasitic corner

   starrc_shell> source <gpd_dir>/starrc_shell_init.tcl

4. Use Parasitic Explorer commands to obtain parasitics for design objects.

starrc_shell> source <gpd_dir>/starrc_shell_load_layout.tcl

5. Use commands to obtain parasitics for design objects.

   starrc_shell> gui_start

```
set gpd_read_remove_buslike_escape false
read_parasitics -keep_capacitive_coupling \
    -format GPD toprt.gpd
current_design toprt
```

<gpd_dir>/starrc_shell_init.tcl

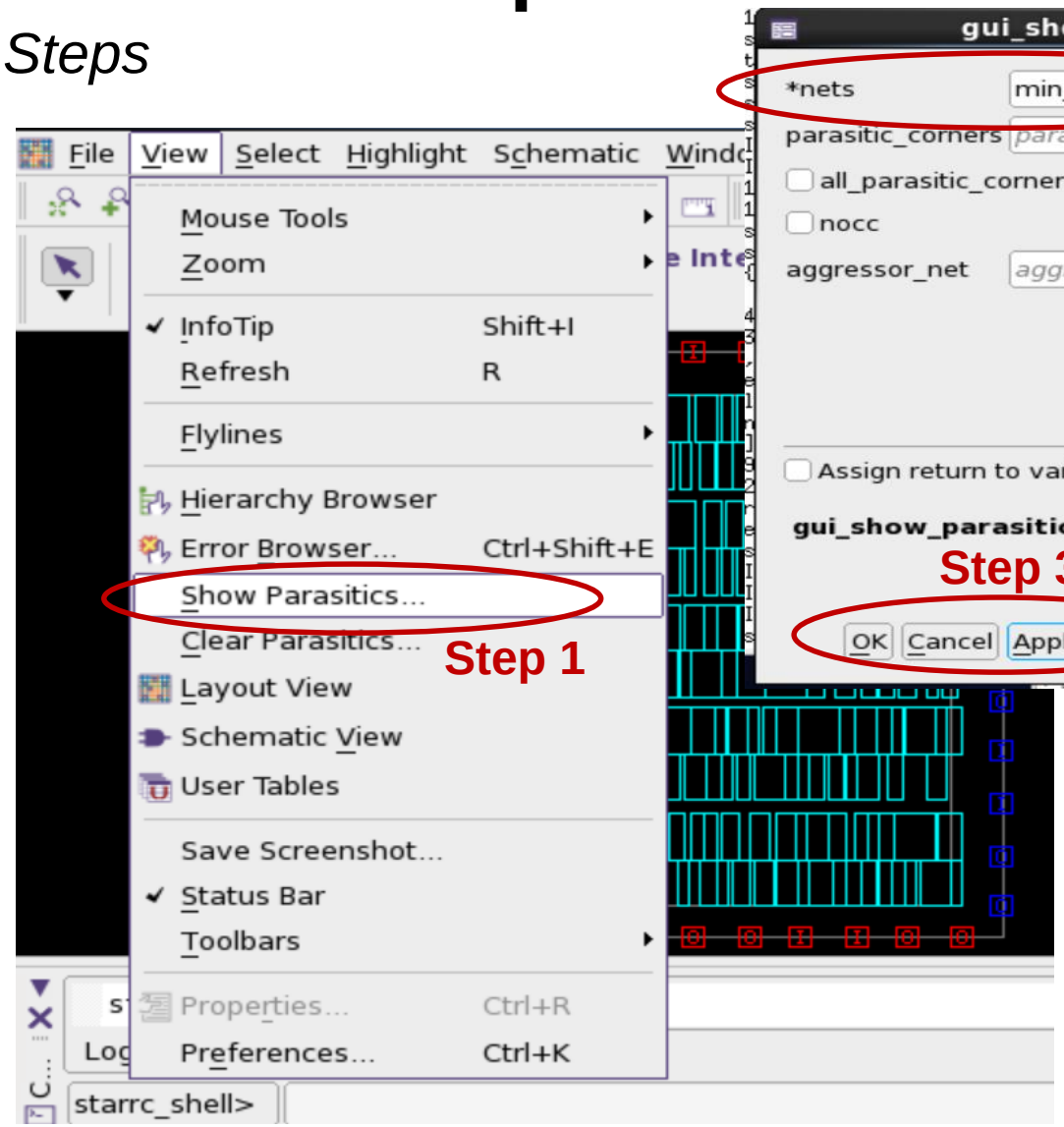
```
set_layout_database_options \
    -physical_enable_clock_data \
    -physical_lib_path { tech.lef cells.lef
    } \
    -physical_design_path { toprt.def }

check_layout_database
```

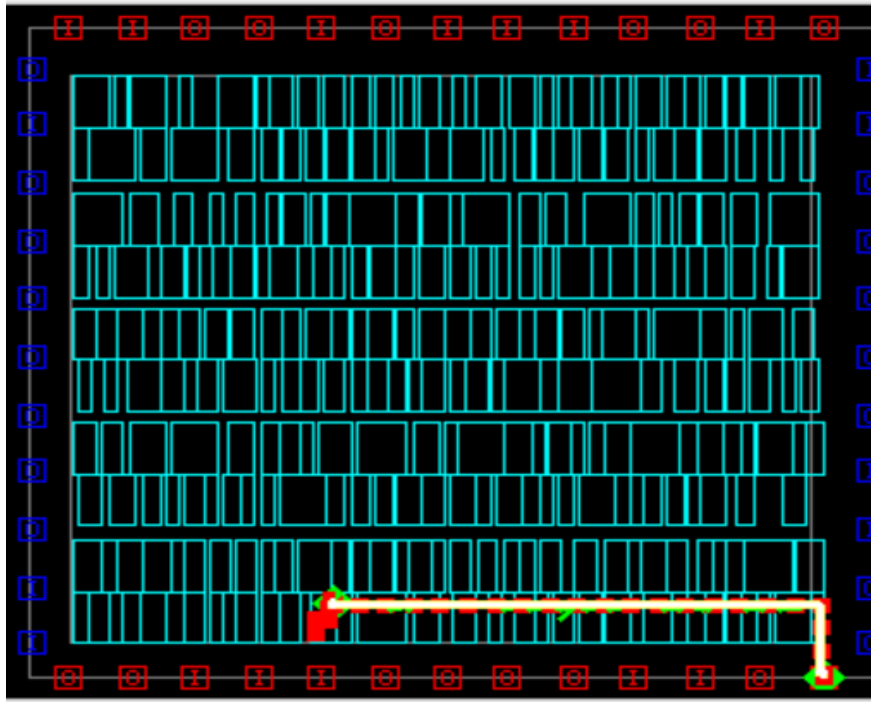
<gpd_dir>/starrc_shell_load_layout.tcl

SYNOPSYS®

Parasitic Explorer GUI – RC Map View

Steps

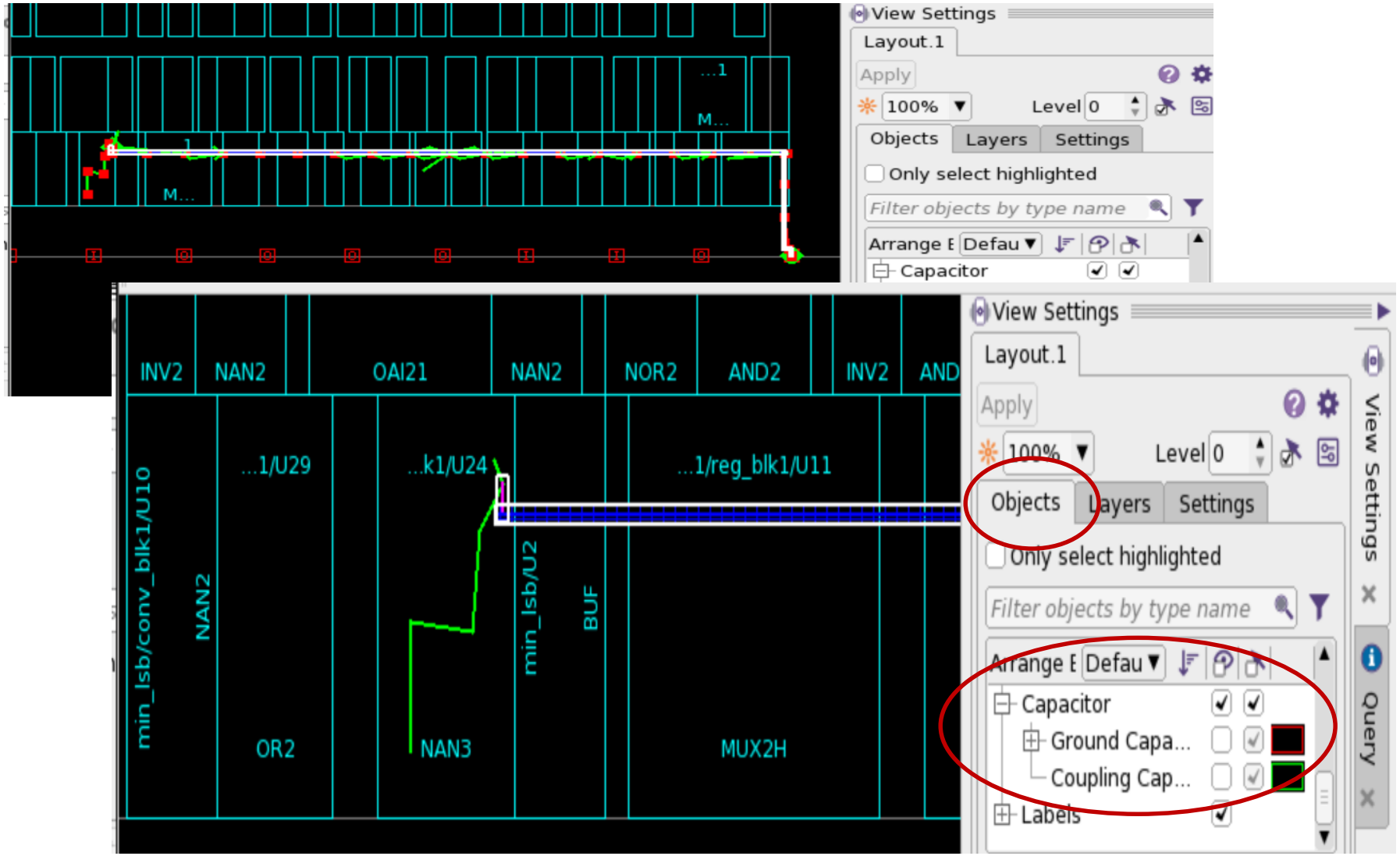




Step 1: Use ‘Show Parasitics’ from ‘View’ menu
Step2 : Specify nets
Step3: Hit ‘Apply’ and net RC is highlighted

SYNOPSYS®

Parasitic Explorer GUI – RC Map View

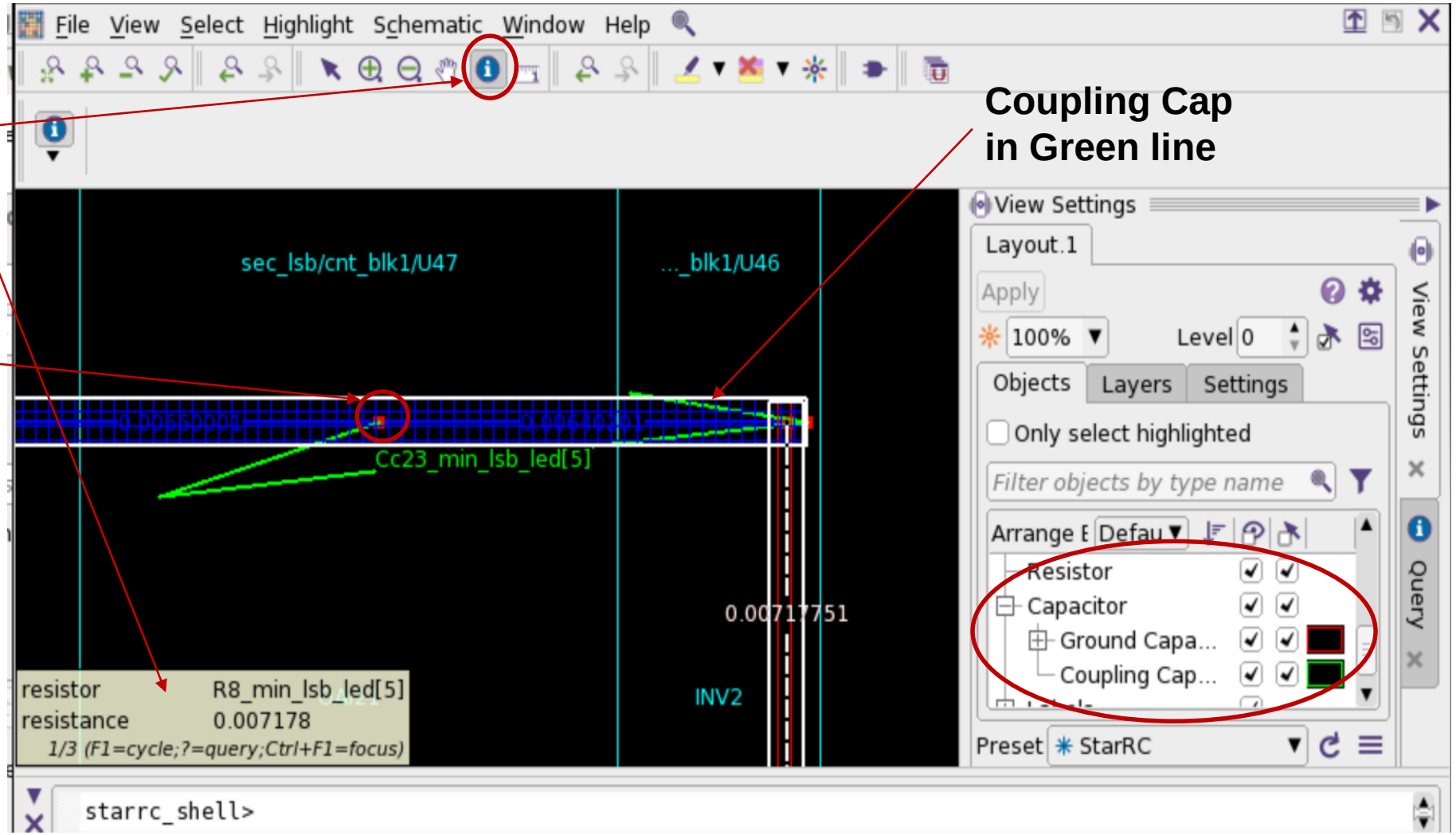


Ground Cap & Coupling Cap Selection from "Object"

SYNOPSYS®

Parasitic Explorer GUI – RC Map Query

Resistor, Ground/Coupling Capacitor & Query

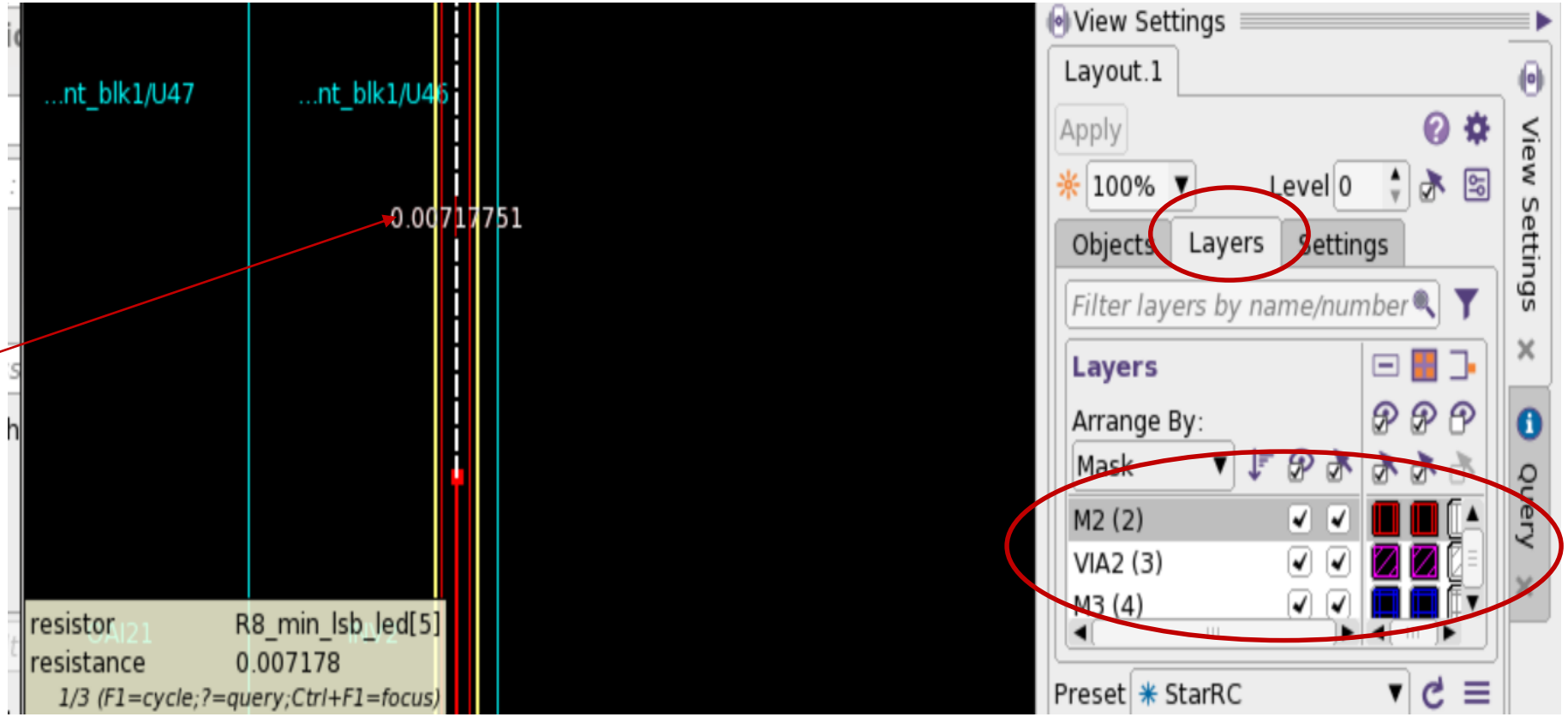


SYNOPSYS®

Parasitic Explorer GUI – RC Map Query

Resistor, Ground/Coupling Capacitor & Query

M2 resistor in Red, highlighted from DEF



Layer Selection

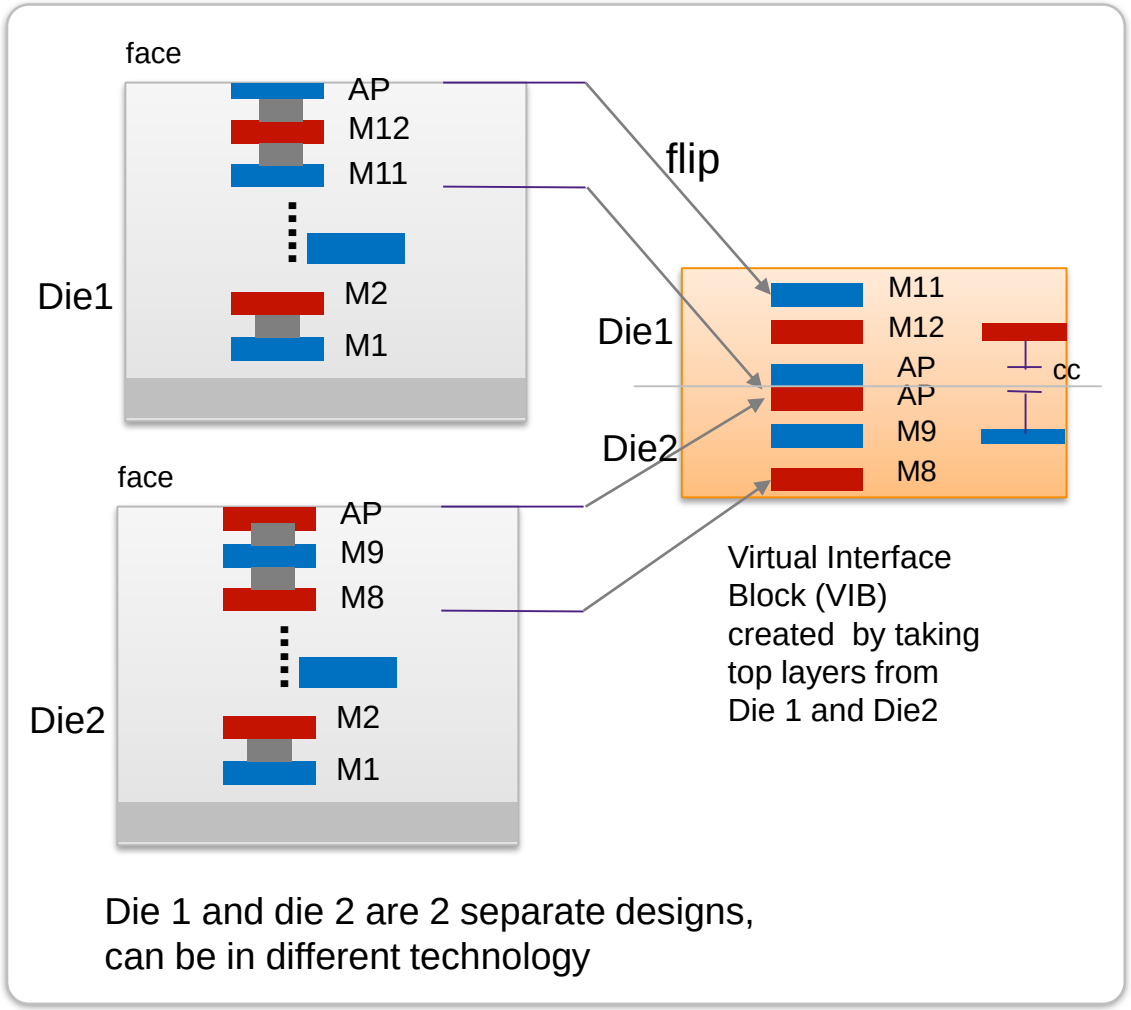
SYNOPSYS®

New Features in Q-2019.12 for Gate-Level Flow

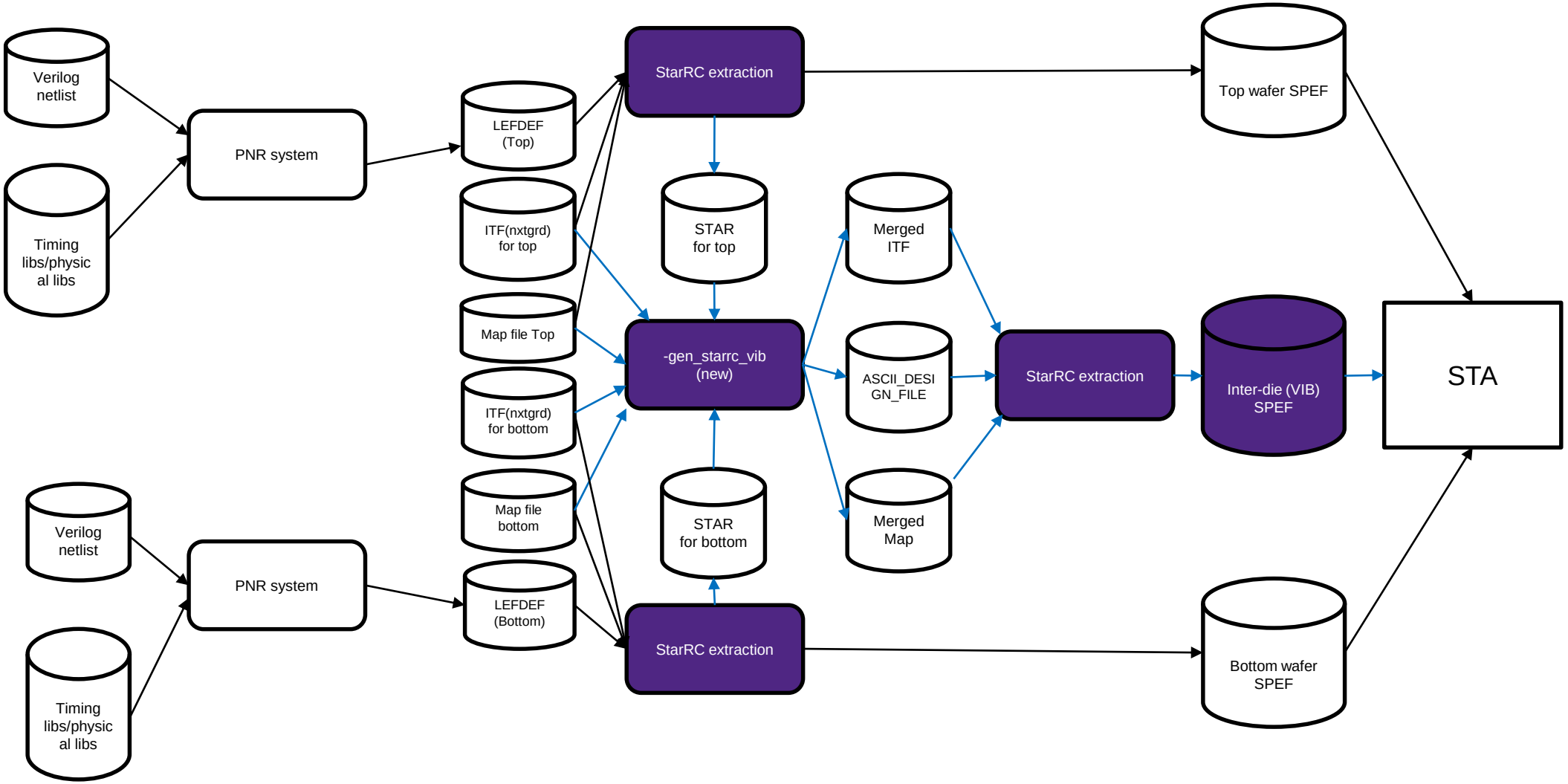
- Gate-Level Performance Improvement
- Virtual Metal Fill Support
- Instance Port Location Closer To The Driver
- Parasitic Explorer –Selected Nets Short Visualization
- Parasitic Explorer – GUI
- LEF/DEF Based 3DIC WoW

StarRC 3DIC WOW LEF/DEF Flow

- In WoW technology, two dies stacked on top of each other are face to face
- Cross-die coupling capacitance on top layers needs to be considered
- Separate virtual interface block (VIB) is created by combining desired top layers from both die
- WoW interface blocks are only used for extraction and DRC/LVS purposes
  - SPEF extracted from virtual interface block is combined with die1 and die2 SPEFs



StarRC 3DIC WOW LEF/DEF Flow



New Features in Q-2019.12 for Tx-Level Flow

- All Angle Extraction On By Default
- Standalone Reducer
- Mutual Inductance Extraction
- Transistor Level Performance Improvement
- Transistor Level GPD Support for Additional Options
- Exploding Trivial Instance Ports
- Parasitic Explorer – Tx Shell
- QuickCap Partition Speed-up

All Angle Extraction On By Default (OBD) for Tx-level Flows

- In P-2019.03 release, a star command **ENHANCED_NONMANHATTAN_ACCURACY** (ENMA) was introduced to improve resistance accuracy for all-angle (including non-Manhattan) shapes. Since then, this feature has been improved both in resistance and capacitance accuracy
- In Q-2019.12 release, ENMA will be on by default. The following results will be expected
  - R-network and resistance accuracy are improved
  - Capacitance accuracy with ENMA on should be comparable to ENMA off

TCAP difference b/t ENMA ON and Off

Designs	Number of nets	Mean Error	StdEV Error
Design1	60	-1.1%	6.5%
Design2	32	-0.60%	0.06%
Design3	12	0.20%	5.3%
Design4	45	-0.93%	0.32%
Design5	15	-2.8%	3.4%

New Features in Q-2019.12 for Tx-Level Flow

- All Angle Extraction On By Default
- Standalone Reducer
- Mutual Inductance Extraction
- Transistor Level Performance Improvement
- Transistor Level GPD Support for Additional Options
- Exploding Trivial Instance Ports
- Parasitic Explorer – Tx Shell
- QuickCap Partition Speed-up

Standalone Reducer

- Features
  - Net based accuracy control for reduction
  - Proven aggressive reduction engine
  - SPF format supported for 2019.12
- User Interface

  StarReduce <reduced command file>
- Wildcards supported for <*net_names*>
- *Default REDUCTION level is HIGH*

Reduced Command File

INPUT_FILENAME : *netlist.spf*
OUTPUT_FILENAME : *output.spf*
REDUCTION_NETS: *<net_names> [LEVEL HIGH|NO] [MAX_DELAY_ERROR <t> | FREQUENCY <v>]*
REDUCTION_NETS: …]
CASE_SENSITIVE: *YES|NO*

Currently available as standalone batch mode

SYNOPSYS®

Standalone Reducer

- Reduced command file and usage example :

red_cmd:
INPUT_FILENAME : xyz.spf
OUTPUT_FILENAME : yyy.spf
REDUCTION_NETS: pll_* LEVEL NO MAX_DELAY_ERROR 1e-12
CASE_SENSITIVE : YES

Usage :
StarReduce red_cmd

New Features in Q-2019.12 for Tx-Level Flow

- All Angle Extraction On By Default
- Standalone Reducer
- Mutual Inductance Extraction
- Transistor Level Performance Improvement
- Transistor Level GPD Support for Additional Options
- Exploding Trivial Instance Ports
- Parasitic Explorer – Tx Shell
- QuickCap Partition Speed-up

Mutual Inductance Extraction

- The new inductance solver is enabled when an **INDUCTANCE_MODE** statement is read from the star_cmd file. Internally to StarRC the following commands are set

REDUCTION: NO

NETLIST_NODE_SECTION: YES

NETLIST_MERGE_SHORTED_PORTS: NO

NETLIST_TAIL_COMMENTS: YES

EXTRA_GEOMETRY_INFO: NODE

PRINT_SILICON_INFO: YES

- StarRC outputs an unreduced netlist containing the physical sizes of resistors (width, thickness, layer) and locations of nodes (x, y, layer).

- When the normal extraction is complete StarRC invokes the stand alone reducer (StarReduce). The main inductance calculation is performed inside the stand alone reducer.

Mutual Inductance Extraction – User Interface

The new inductance capability is enabled when an INDUCTANCE_MODE statement is read from star_cmd. The following new commands have been added to STAR:

INDUCTANCE_MODE: <INDUCTANCE | RELUCTANCE>

Enables new inductance functionality and selects inductance or reluctance for output.

Default: No inductance calculation

INDUCTANCE_SELECT_LAYER: <layer1_name> <layer2_name> …

Selects layers for inductance extraction

Default: all layers

INDUCTANCE_SELECT_NET: <net1_name> <net2_name> …

Selects layers for inductance extraction

Default: all nets

Mutual Inductance Extraction – User Interface

INDUCTANCE_SELECT_BB: (float_xl float_yl float_xh float_yh <layer_1> <layer_2> ….) | (FILE file_name)

Specifies a bounding box for inductance extraction. Units are microns.

The FILE option specifies the name of a file containing one or more selection boxes.

Syntax:

float_xl float_yl float_xh float_yh <layer_1> <layer_2> ….

Default: the whole design

INDUCTANCE_FREQUENCY: float

Specifies the frequency at which inductance is calculated. Specifying a higher frequency produces a smaller skin depth so a finer discretization is needed (INDUCTANCE_NINC is automatically increased) and the run time becomes longer. Units are GHz

Default: 1

Mutual Inductance Extraction – User Interface

INDUCTANCE_NINC: int

Specifies the number of filaments to divide each metal segment into for skin depth and proximity effect calculation.

Increasing this parameter will increase accuracy but strongly increase runtime and strongly reduce capacity. Suggest leaving it at its default.

Default: Automatic

INDUCTANCE_MIN_LENGTH: float

Specifies the minimum length for a resistor (metal segment) to be included in inductance.

Default: 1um

INDUCTANCE_REL_THRESHOLD: float_percent

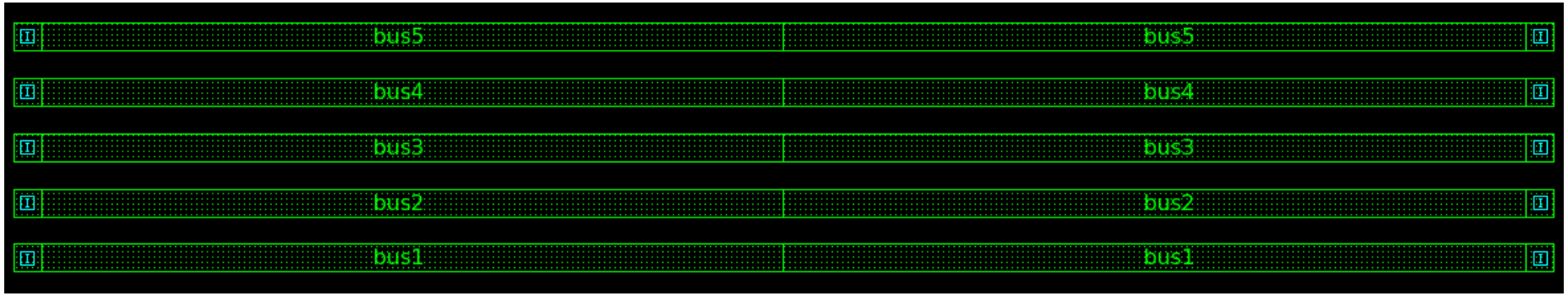
Reluctance terms less than this percentage of the corresponding matrix diagonal are dropped. Increasing this parameter reduces accuracy and simulation time.

Default: 1 percent.

Mutual Inductance Test Results

Standalone Fast Henry vs StarRC on Parallel Bus (Manual Testcase)

- Comparison of Impedance values between Fast Henry and StarRC
- Design Spec:
  - 5 parallel lines
  - 20um long, 0.36um wide, 0.85 thick (Minimum values)
  - Spacing between each bus is 0.36um.
  - Etching is off.
  - Using TSMC 7nm nxtgrd file.
  - Frequency: 2GHz



SYNOPSYS®

Test Results

Standalone Fast Henry vs StarRC on Parallel Bus (Manual Testcase)

- Comparison of Impedance values between Fast Henry and StarRC

	StarRC	FH	%Diff w.r.t. FH
R0=R1=R2=R3=R4	1.37	1.39	-1.438848921
L0 (Bus1)	15.785	16.046	-1.626573601
L2 (Bus2)	15.78	16.041	-1.627080606
L4 (Bus3)	15.777	16.038	-1.627384961
L6 (Bus4)	15.776	16.037	-1.627486438
L8 (Bus5)	15.776	16.037	-1.627486438
K_0 L0 L2	0.740921	0.74142	-0.067303283
K_1 L0 L4	0.590388	0.591194	-0.136334266
K_2 L0 L6	0.500611	0.501628	-0.202739879
K_3 L0 L8	0.438257	0.439461	-0.273971979
K_4 L2 L4	0.740861	0.741494	-0.085368189
K_5 L2 L6	0.590336	0.591327	-0.167589168
K_6 L2 L8	0.500589	0.501805	-0.242325206
K_7 L4 L6	0.740835	0.741558	-0.097497431
K_8 L4 L8	0.59032	0.591442	-0.189705838
K_9 L6 L8	0.740829	0.741613	-0.105715515

SYNOPSYS®

Mutual Inductance Extraction –Set-up (Example)

- Expected command file setup:

INDUCTANCE_MODE: INDUCTANCE

INDUCTANCE_SELECT_BB: -36.5 12 -6.4 43.3

INDUCTANCE_SELECT_LAYER: M13 M12 M11

INDUCTANCE_REL_THRESHOLD: 1

*INDUCTANCE_MIN_LENGTH: 0.7

INDUCTANCE_SELECT_NET: vo_m

INDUCTANCE_FREQUENCY: 1

- Recommended Settings:

POWER_EXTRACT: YES

MERGE_VIAS_IN_ARRAY: YES

New Features in Q-2019.12 for Tx-Level Flow

- All Angle Extraction On By Default
- Standalone Reducer
- Mutual Inductance Extraction
- Transistor Level Performance Improvement
- Transistor Level GPD Support for Additional Options
- Exploding Trivial Instance Ports
- Parasitic Explorer – Tx Shell
- QuickCap Partition Speed-up

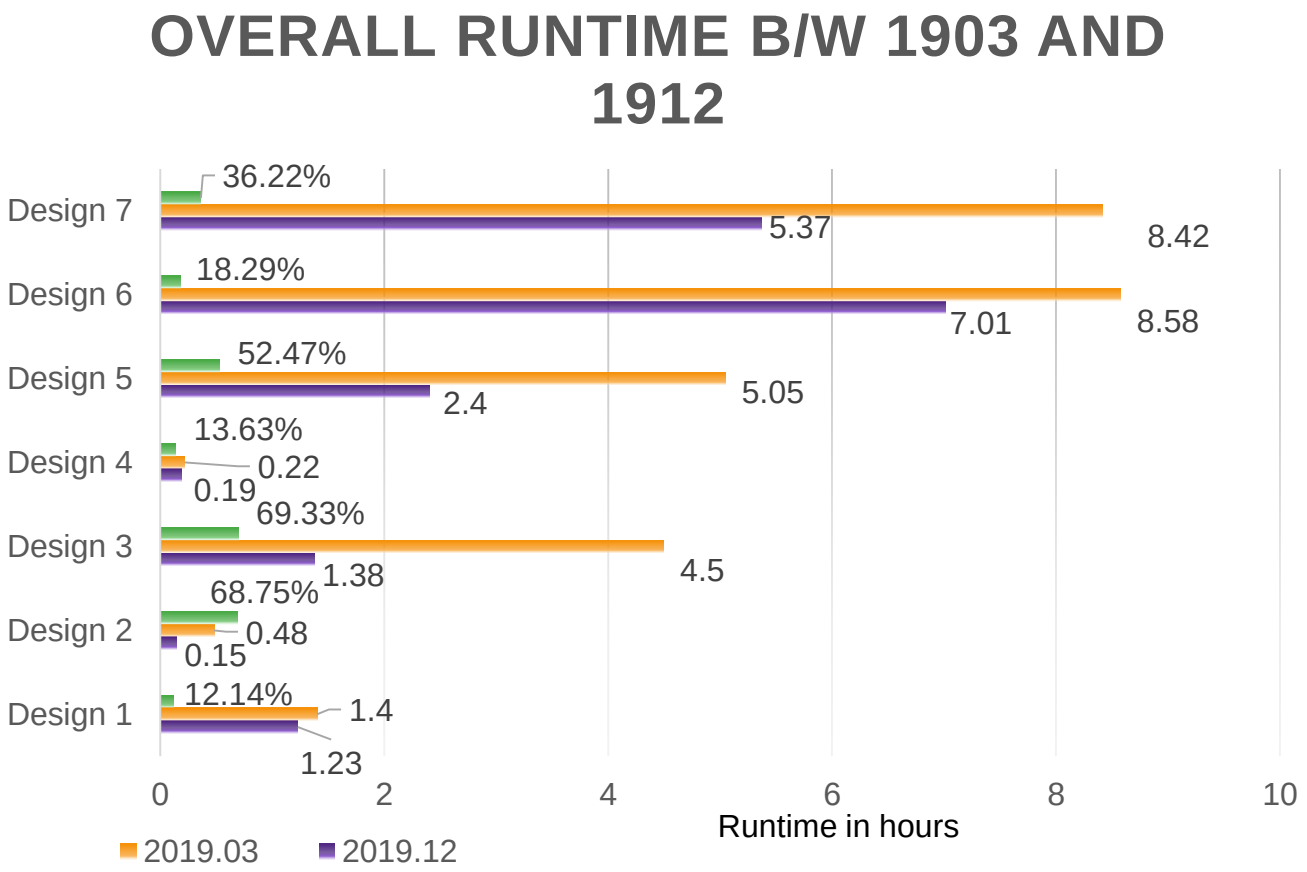
Transistor level performance improvement

- 1.5X faster runtime VS P-2019.03
  - Single corner run
  - GPD enabled
  - 8 cores
  - Higher speedup when reduction: high/netlist_remove_dangling_branches: yes/mos_gate_non_negative_delta_resistance:yes is set
  - Similar accuracy to previous release

- Source of speedup
  - Netlist stage DP (good scalability when using 1-8 cores)
  - Other enhancements:
    - Algorithm optimizations for netlist_remove_dangling_branches/mos_gate_non_negative_delta_resistance/opens detection/xref:yes
    - Via merge stage DP

Tx-level Performance Improvement

- Improvements in pre-extract and post-extract stages
  - **Pre-Extract Stage**
    - Distributed processing for ViaMerge stage
    - Performance improvement in Xref and HN stages
    - DP for TRBL:CONLY and pcell layers file
  - **Post-Extract Stage**
    - Memory and performance optimizations in GPD flow
    - Netlist DP feature in GPD flow
  - Average percentage improvement : 38%



SYNOPSYS®

New Features in Q-2019.12 for Tx-Level Flow

- All Angle Extraction On By Default
- Standalone Reducer
- Mutual Inductance Extraction
- Transistor Level Performance Improvement
- Transistor Level GPD Support for Additional Options
- Exploding Trivial Instance Ports
- Parasitic Explorer – Tx Shell
- QuickCap Partition Speed-up

Tx GPD 2nd Tier Options Support

- List of Previously unsupported commands that will be supported in GPD for 2019.12
  - Full VI/OA flow support (OBSERVATION_POINTS support for probing available in 2019.12 production code)
  - Support FS flow
  - NETLIST_FORMAT: NETNAME
  - NETLIST_NODENAME_NETNAME support on dangling ports (new command NODENAME_NETNAME_ON_DANGLING_PORTS will be added)
  - CAPACITOR_TAIL_COMMENTS
  - EM_PARAM_MAPPING_FILE
  - EXPLODE_TRIVIAL_INSTANCE_PORTS
  - NETLIST_COUPLE_UNSELECTED_NETS:IDEAL
  - NETLIST_SELECT_NETS with netlist reduction
  - EXTRACTION: R
  - NETLIST_SWAP_TERMINAL
  - NETLIST_MOVE_SPICE_TYPE_TO_LAST

Tx GPD on by Default since 2019.03

Tx GPD New Features

- Enhance GPD to have more information:
  - –Scale information
  - –Prefix for layout only devices and layout only nets
  - –Text file to know INSTANCE SECTION

- Benefits
  - –To improve performance and
  - –To improve EOU for BA flow
  - –Supported by FineSim 2019.06 or later versions.

New Features in Q-2019.12 for Tx-Level Flow

- All Angle Extraction On By Default
- Standalone Reducer
- Mutual Inductance Extraction
- Transistor Level Performance Improvement
- Transistor Level GPD Support for Additional Options
- Exploding Trivial Instance Ports
- Parasitic Explorer – Tx Shell
- QuickCap Partition Speed-up

ln_* instance port explode option

- Sometimes there are untexted ports that are generated during LVS.
- These are also referred to as trivial instance ports
- Such ports are created during LVS due to non-critical material (such as fill) overlap between skip cell and the parent.
- These are not connected down to any devices in the hierarchy.
- These instance ports come out with a ln_ prefix by default in StarRC output.
- These ports do not exist in the schematic/verilog, and can cause back annotation issues during STA.

ln_* instance port explode option

- To avoid issues downstream, users would prefer a way to explode these ports.

- StarRC is enhanced with new command

  `EXPLODE_TRIVIAL_INSTANCE_PORTS:YES` (default:no)

  to support this feature

- This removes the ln_ports from the netlist, by exploding them to the parent level.

- It will also remove the trivial ports in NETLIST_IDEAL_SPICE_FILE (exception is when SPICE_SUBCKT_FILE has port definitions for the SKIP_CELLS, in which case the ports are dictated by SPICE_SUBCKT_FILE.)

New Features in Q-2019.12 for Tx-Level Flow

- All Angle Extraction On By Default
- Standalone Reducer
- Mutual Inductance Extraction
- Transistor Level Performance Improvement
- Transistor Level GPD Support for Additional Options
- Exploding Trivial Instance Ports
- Parasitic Explorer – Tx Shell
- QuickCap Partition Speed-up

Parasitic Explorer – Extended to Tx-level

Interactive Design Analysis and Exploration

Done
WIP
Planned

- *Dominant layer in a timing path*
- *Identify nets contributing higher RC on a path*
- *Net with dominating RC on a timing path*
- *Worst aggressor on a net in timing path*
- *Average aggressor on a net in timing path*

Path Based Analysis

Net Based Analysis

- *Layer wise length distribution of a net*
- *Identify nets routed on top metal layers*
- *Total Cap, Ground Cap, Resistance report*
- *Report width/layer of all resistor segments of a net*
- *Point to point equivalent resistance between driver and receiver*
- Identify shortest resistive path on a net
- List all non-physical resistors for EM

- *Capacitance variation between specific corners*
- Resistance variation between specific corners

Corner Based Analysis

GUI Support Under Planning

- Use Virtuoso Interface as UI for VI customers
- Use CustomCompiler as UI for CC customers

SYNOPSYS®

Tcl Flow and Usage in starrc_shell

Usage

1. Run extraction with the following command

   ```
   PARASITIC_EXPLORER_ENABLE_ANALYSIS: YES
   ```

2. Invoke the StarRC shell

   ```
   % starrc_shell
   ```

3. For SMC designs, set the parasitic corner

   ```
   starrc_shell> source  <gpd_dir>/starrc_shell_init.tcl
   ```

4. Use Parasitic Explorer commands to obtain parasitics for design objects.

   ```
   starrc_shell> get_coupling_capacitors ...
   starrc_shell> get_ground_capacitors ...
   starrc_shell> get_resistors ...
   ```

```
set gpd_read_remove_buslike_escape false
read_parasitics -keep_capacitive_coupling \
    -format GPD toprt.gpd
current_design toprt
```

<gpd_dir>/starrc_shell_init.tcl

Recap: Parasitic Explorer Command Examples

Coupling Capacitor Attributes

Report coupling capacitors of a net

```
starrc_shell> report_attribute -application [get_coupling_capacitors -of_objects min_lsb/n76]
```

Output (truncated)

```
****************************************
Report : Attribute
Design : top_test_macro
Version: P-2019.03
Date   : Mon Mar 18 17:13:04 2019
****************************************
Design          Object             Type       Attribute Name          Value
-------------------------------------------------------------------------------
top_test_macro  coupling_capacitor collection aggressor_net           sec_msb/n67
top_test_macro  coupling_capacitor string     aggressor_net_name      sec_msb/n67
top_test_macro  coupling_capacitor collection aggressor_node_ground_capacitor ground_capacitor
top_test_macro  coupling_capacitor int        aggressor_node_index    6
top_test_macro  coupling_capacitor string     aggressor_node_name     sec_msb/n67:6
top_test_macro  coupling_capacitor float      capacitance             0.000120
top_test_macro  coupling_capacitor int        layer_id                6
top_test_macro  coupling_capacitor string     layer_name              M6
top_test_macro  coupling_capacitor collection net                     min_lsb/n76
top_test_macro  coupling_capacitor collection node_ground_capacitor   ground_capacitor
top_test_macro  coupling_capacitor int        node_index              9
top_test_macro  coupling_capacitor string     node_name               min_lsb/n76:9
top_test_macro  coupling_capacitor string     object_class            coupling_capacitor
```

Corresponding SPEF Cc

```
*NAME_MAP
*352 min_lsb/n76
*843 sec_msb/n67

*D_NET *352 24.6342
...

*CONN
...

*CAP
...
4 *352:8 *843:5 0.120283
...

*RES
...
```


Resistor Attributes

Report resistors of a net

```
starrc_shell> report_attribute -application [get_resistors -of_objects min_lsb/n76]
```

Output

```
Design          Object      Type       Attribute Name          Value
-------------------------------------------------------------------------------
top_test_macro  resistor    boolean    is_short                false
top_test_macro  resistor    int        layer_id                2
top_test_macro  resistor    string     layer_name              M1
top_test_macro  resistor    collection net                     min_lsb/n76
top_test_macro  resistor    collection node1_ground_capacitor  ground_capacitor
top_test_macro  resistor    int        node1_index             13
top_test_macro  resistor    string     node1_name              min_lsb/n76:13
top_test_macro  resistor    collection node2_ground_capacitor  ground_capacitor
top_test_macro  resistor    int        node2_index             6
top_test_macro  resistor    string     node2_name              min_lsb/n76:6
top_test_macro  resistor    string     object_class            resistor
top_test_macro  resistor    float      resistance              0.010415
```

Corresponding SPEF Resistor

```
*NAME_MAP
*352 min_lsb/n76

*D_NET *352 24.6342
...

*CONN
...

*CAP
...

*RES
...
7 *352:12 *352:5 10.4149
...
```


Ground Capacitor Attributes

Report ground capacitors of a net

```
starrc_shell> report_attribute -application [get_ground_capacitors -of_objects min_lsb/n76]
```

Output

```
Design          Object             Type       Attribute Name          Value
-------------------------------------------------------------------------------
top_test_macro  ground_capacitor   float      capacitance             0.000511
top_test_macro  ground_capacitor   int        layer_id                2
top_test_macro  ground_capacitor   string     layer_name              M2
top_test_macro  ground_capacitor   collection net                     min_lsb/n76
top_test_macro  ground_capacitor   int        node_index              7
top_test_macro  ground_capacitor   string     node_name               min_lsb/n76:7
top_test_macro  ground_capacitor   string     node_type               internal_node
top_test_macro  ground_capacitor   string     object_class            ground_capacitor
top_test_macro  ground_capacitor   float      x_coordinate_center     373.399994
top_test_macro  ground_capacitor   float      x_coordinate_max        373.489990
top_test_macro  ground_capacitor   float      x_coordinate_min        373.309998
top_test_macro  ground_capacitor   float      y_coordinate_center     324.399994
top_test_macro  ground_capacitor   float      y_coordinate_max        324.450012
top_test_macro  ground_capacitor   float      y_coordinate_min        324.140015
```

Corresponding SPEF Cg

```
*NAME_MAP
*352 min_lsb/n76

*D_NET *352 24.6342
...

*CONN
...

*CAP
...
33 *352:6 0.510786
...

*RES
...
```


Multicorner GPD Usage Model

- A GPD may consist of simultaneous multicorner extraction data
- Parasitic explorer allows access to data from these multiple corners
- You can query RC elements for all corners or a subset of corners available in the GPD

```
starrc_shell> report_attribute -application [get_resistors -of_objects min_lsb/n76]

Design          Object      Type       Attribute Name          Value
-------------------------------------------------------------------------------
top_test_macro  resistor    int        node1_index             13
top_test_macro  resistor    string     node1_name              min_lsb/n76:13
top_test_macro  resistor    collection node2_ground_capacitor  ground_capacitor
top_test_macro  resistor    int        node2_index             6
top_test_macro  resistor    string     node2_name              min_lsb/n76:6
top_test_macro  resistor    string     object_class            resistor
top_test_macro  resistor    float      resistance              0.0104149
```

```
starrc_shell> report_attribute -application [get_resistors -of_objects min_lsb/n76 -
all_parasitic_corners]

Design          Object      Type       Attribute Name          Value
-------------------------------------------------------------------------------
top_test_macro  resistor    int        node1_index             13
top_test_macro  resistor    string     node1_name              min_lsb/n76:13
top_test_macro  resistor    collection node2_ground_capacitor  ground_capacitor
top_test_macro  resistor    int        node2_index             6
top_test_macro  resistor    string     node2_name              min_lsb/n76:6
top_test_macro  resistor    string     object_class            resistor
top_test_macro  resistor    float      resistance_multicorner  0.0104149 0.0108459 0.0103234
```


SYNOPSYS®

Recap: Parasitic Explorer Point to Point Resistance

Report the equivalent or effective resistance between two nodes of a net

```
starrc_shell> get_point_to_point_resistance -from min_msb/U21/A -to min_msb/U20/X
0.0176175
```

Syntax

`get_point_to_point_resistance`	
`[-quiet]`	Suppresses warning and error messages no objects exist
`[-parasitic_corners corner_names]`	Specifies the corners in the GPD to query (for SMC GPD)
`[-all_parasitic_corners]`	Queries all parasitic corners (for SMC GPD)
`-from node1 -to node2`	Specifies the start and end point. Can be a pin, port or internal node

User TCL: report_point_to_point_resistance

Reports P2P R for all pin/instance port combinations of the specified net

Usage:

starrc_shell> report_point_to_point_resistance \

-of_objects <list or collection of nets>

Output:

- Pin1, pin2 and equivalent P2P R
- All combinations of *P/*I are computed

```
starrc_shell> report_point_to_point_resistance -of_objects "SUM0 B0"
========================
Net: SUM0
Report Type: P2P R
========================
Pin1       Pin2       P2P R
====       ====       =====
SUM0       0/33/M2/s  0.002518
SUM0       0/33/M1/s  0.002802
0/33/M2/s  0/33/M1/s  0.001385
========================
Net: B0
Report Type: P2P R
========================
Pin1       Pin2       P2P R
====       ====       =====
B0         0/38/M5/g  0.051076
B0         0/38/M2/g  0.027876
B0         0/53/M3/g  0.059847
B0         0/53/M1/g  0.036113
B0         0/49/M3/g  0.046901
B0         0/49/M1/g  0.043435
B0         0/54/M5/g  0.043880
B0         0/54/M2/g  0.043079
0/38/M5/g  0/38/M2/g  0.074400
0/38/M5/g  0/53/M3/g  0.108194
0/38/M5/g  0/53/M1/g  0.084461
0/38/M5/g  0/49/M3/g  0.095249
```

Output

Recap: Parasitic Explorer RC Object Attributes

```
starrc_shell> list_attributes -application -class resistor

****************************************

Attribute Name                Object     Type       Properties
-------------------------------------------------------------
area                          resistor   float      A
is_short                      resistor   boolean    A
is_via                        resistor   boolean    A
is_via_array                  resistor   boolean    A
is_via_ladder_em              resistor   boolean    A
is_via_ladder_high_performance resistor boolean A
layer_id                      resistor   int        A
layer_name                    resistor   string     A
length                        resistor   float      A
net                           resistor   collection A
node1_ground_capacitor        resistor   collection A
node1_index                   resistor   int        A
node1_name                    resistor   string     A
node2_ground_capacitor        resistor   collection A
node2_index                   resistor   int        A
node2_name                    resistor   string     A
object_class                  resistor   string     A
resistance                    resistor   float      A
resistance_max                resistor   float      A
resistance_min                resistor   float      A
resistance_multicorner        resistor   string     A
via_array_nx                  resistor   int        A
via_array_ny                  resistor   int        A
via_array_perimeter           resistor   float      A
width                         resistor   float      A
x_coordinate_max              resistor   float      A
x_coordinate_min              resistor   float      A
y_coordinate_max              resistor   float      A
y_coordinate_min              resistor   float      A
```

Some attribute definitions depend on extraction settings

StarRC Setting	Associated Attributes
NETLIST_TAIL_COMMENTS: YES	• is_via • is_via_array • length • width
EXTRA_GEOMETRY_INFO: RES	• x_coordinate_max • x_coordinate_min • y_coordinate_max • y_coordinate_min
Single corner extraction	• resistance
SMC extraction	• resistance_max • resistance_min resistance_multicorner

Tx-level: report_coupling_capacitors

Report coupling capacitors of a net

```
starrc_shell> source public/proc_report_coupling_capacitors.tcl
starrc_shell> report_coupling_capacitors -of_objects SUM0
============================
Net: SUM0
Total capacitance: 0.013721
Report Type: Aggressors, summary
============================
Total CCAP      %Cc/Ct          Aggressor Net
==========      =====           =============
0.000925        6.741491        B0
0.000908        6.617593        A0
0.000468        3.410830        CIN
starrc_shell> get_pins -of [get_nets SUM0]
{"0/33/M2/s", "0/33/M1/s"}
starrc_shell> report_coupling_capacitors -of_objects SUM0 -verbose
============================
Net: SUM0
Total capacitance: 0.013721
Report Type: Aggressors, detailed
============================
Victim Node Victim Layer Aggressor Node Aggressor Layer Capacitance %Cc/Ct
=========== ============ ============== ============== =========== ========
0/33/M2/s   SUBSTRATE    A0:24          metal1          0.000299    2.179141
SUM0:5      metal2       A0:24          metal1          0.000047    0.342541
0/33/M2/s   SUBSTRATE    A0:25          metal1          0.000060    0.437286
SUM0:5      metal2       A0:25          metal1          0.000502    3.658625
0/33/M2/s   SUBSTRATE    B0:25          metal1          0.000370    2.696596
SUM0:4      metal2       B0:25          metal1          0.000001    0.007288
SUM0:5      metal2       B0:25          metal1          0.000504    3.673202
SUM0:5      metal2       B0:26          metal1          0.000050    0.364405
0/33/M1/s   SUBSTRATE    CIN:20         metal1          0.000354    2.579987
SUM0:9      metal1       CIN:20         metal1          0.000109    0.794403
SUM0:9      metal1       CIN:21         metal1          0.000005    0.036440
```

```
* Instance Section
*
M0/38/M3 0/38/M3:d 0/38/M3:g GND GND n AD=19.5p AS=39p L=1u PD=16u PS=32u W=13u
M0/38/M2 0/38/M2:d 0/38/M2:g 0/38/M2:s GND n AD=19.5p AS=19.5p L=1u PD=16u PS=16u W=13u
M0/39/M1 GND 0/39/M1:g 0/39/M1:s GND n AD=39p AS=39p L=1u PD=32u PS=32u W=13u
M3/52/M1 GND 3/52/M1:g 3/52/M1:s GND n AD=39p AS=39p L=1u PD=32u PS=32u W=13u
M3/53/M1 3/53/M1:d 3/53/M1:g 3/53/M1:s GND n AD=13p AS=39p L=1u PD=15u PS=32u W=13u
```

```
starrc_shell> get_cells 0/38/M3
{"0/38/M3"}
starrc_shell> get_pins -of [get_cells 0/38/M3]
{"0/38/M3/d", "0/38/M3/g", "0/38/M3/s", "0/38/M3/b"}
```

```
starrc_shell> get_nets -of [get_pins 0/38/M3/g]
{"A0"}
starrc_shell> report_coupling_capacitors -of_objects A0
============================
Net: A0
Total capacitance: 0.092668
Report Type: Aggressors, summary
============================
Total CCAP      %Cc/Ct          Aggressor Net
==========      =====           =============
0.014877        16.054086       B0
0.000908        0.979842        SUM0
starrc_shell> report_coupling_capacitors -of_objects A0 -verbose
============================
Net: A0
Total capacitance: 0.092668
Report Type: Aggressors, detailed
============================
Victim Node Victim Layer Aggressor Node Aggressor Layer Capacitance %Cc/Ct
=========== ============ ============== ============== =========== ========
A0:10       metal2       B0:10          metal2          0.000060    0.064747
A0:10       metal2       B0:11          metal2          0.000008    0.008633
A0:10       metal2       B0:22          metal1          0.000018    0.019424
A0:11       metal2       B0:13          metal2          0.000039    0.042086
A0:12       metal2       B0:11          metal2          0.000148    0.159710
A0:12       metal2       B0:12          metal2          0.000042    0.045323
A0:12       metal2       B0:13          metal2          0.000011    0.011870
A0:12       metal2       B0:23          metal1          0.000004    0.004316
A0:12       metal2       B0:24          metal1          0.000004    0.004316
A0:13       metal2       B0:13          metal2          0.000192    0.207191
```


SYNOPSYS®

Tx-level: report_ground_capacitors

Report ground capacitors of a net

Usage:

starrc_shell> report_ground_capacitors \

[-of_objects “list or collection of nets”]

[-from “from pin/port/node”]

[-to “to pin/port/node”]

For nets with escape character, use “get_nets -exact”

e.g. report_ground_capacitors –of_objects [get_nets -exact {net\\[0\\]}]

Output:

- Report with nodes, layer, capacitance, %Cg/Ct
- Report of all ground capacitors of a net along with % ratio of ground capacitance to total capacitance

```
starrc_shell> report_ground_capacitors  -of_objects “SUM0 B0”
==============================
Net: SUM0
Total capacitance: 0.013721
Report Type: Ground Capacitors
==============================
Node       Layer     Capacitance %Cg/Ct
====       =====     =========== ========
0/33/M2/s  SUBSTRATE 0.000749    5.458786
0/33/M1/s  SUBSTRATE 0.000387    2.820494
SUM0:4     metal2    0.000247    1.800160
SUM0:5     metal2    0.002221    16.186867
SUM0:6     metal2    0.002150    15.669412
SUM0:7     metal2    0.001408    10.261643
SUM0:8     metal1    0.001582    11.529772
SUM0:9     metal1    0.000479    3.490999
SUM0:10    metal1    0.000255    1.858465
SUM0:11    metal1    0.000293    2.135413
SUM0:12    metal1    0.001649    12.018074
==============================
Net: B0
Total capacitance: 0.089779
Report Type: Ground Capacitors
==============================
Node       Layer  Capacitance %Cg/Ct
====       =====  =========== ========
B0         metal2 0.000000    0.000000
0/38/M2/g  poly   0.000000    0.000000
0/38/M5/g  poly   0.000000    0.000000
0/54/M5/g  poly   0.000000    0.000000
0/49/M3/g  poly   0.000000    0.000000
0/54/M2/g  poly   0.000000    0.000000
0/53/M3/g  poly   0.000000    0.000000
0/49/M1/g  poly   0.000000    0.000000
0/53/M1/g  poly   0.000000    0.000000
B0:10      metal2 0.000082    0.091335
B0:11      metal2 0.001010    1.124985
```

Ground Capacitors Report

SYNOPSYS®

Tx-level: report_resistors

Report resistors of a net

Usage:

starrc_shell> report_resistors \

[-of_objects “list or collection of nets”]

[-from “from pin/port/node”]

[-to “to pin/port/node”]

[-verbose]

For nets with escape character, use “get_nets –exact”

e.g. report_resistors -of_objects [get_nets -exact {net\\[0\\]}]

Output:

- Net or path based report with nodes, resistance, layer, length, width, bbox
- Default “-type” is “summary”. Reports only nodes and resistance value
- Detailed report lists layer, length, width, bbox

```
starrc_shell> report_resistors -of_objects “SUM0 B0”
============================
Net: SUM0
Report Type: Resistors, Net Based, summary
============================
Node1      Node2    Resistance
=====      =====    ==========
SUM0       SUM0:6   0.000357
SUM0       SUM0:7   0.000031
0/33/M2/s  SUM0:11  0.000550
0/33/M1/s  SUM0:15  0.000550
SUM0:4     SUM0:5   0.000620
SUM0:4     SUM0:13  0.000031
SUM0:5     SUM0:6   0.000620
SUM0:8     SUM0:10  0.000106
SUM0:8     SUM0:11  0.000085
SUM0:9     SUM0:10  0.000062
```

Net Based Summary Report

```
starrc_shell> report_resistors -from 0/33/M2/s -to SUM0:12
============================
Net: SUM0
From: 0/33/M2/s
To: SUM0:12
Report Type: Resistors, Path Based, summary
============================
Node1      Node2    Resistance
=====      =====    ==========
0/33/M1/s  SUM0:15  0.000550
SUM0:12    SUM0:14  0.000031
SUM0:13    SUM0:14  0.000237
SUM0:13    SUM0:16  0.000031
```

Path Based Summary Report

```
starrc_shell> report_resistors -of SUM0 -verbose
============================
Net: SUM0
Report Type: Resistors, Net Based, detailed
============================
Node1      Node2    Resistance  Layer      Length Width Area llx lly urx ury
=====      =====    ==========  =====      ====== ===== ==== === === === ===
SUM0       SUM0:6   0.000357    metal2     NA     NA    NA   NA  NA  NA  NA
SUM0       SUM0:7   0.000031    metal2     NA     NA    NA   NA  NA  NA  NA
0/33/M2/s  SUM0:11  0.000550    SUBSTRATE  NA     NA    NA   NA  NA  NA  NA
0/33/M1/s  SUM0:15  0.000550    SUBSTRATE  NA     NA    NA   NA  NA  NA  NA
SUM0:4     SUM0:5   0.000620    metal2     NA     NA    NA   NA  NA  NA  NA
SUM0:4     SUM0:13  0.000031    metal2     NA     NA    NA   NA  NA  NA  NA
SUM0:5     SUM0:6   0.000620    metal2     NA     NA    NA   NA  NA  NA  NA
SUM0:8     SUM0:10  0.000106    metal1     NA     NA    NA   NA  NA  NA  NA
SUM0:8     SUM0:11  0.000085    metal1     NA     NA    NA   NA  NA  NA  NA
SUM0:9     SUM0:10  0.000062    metal1     NA     NA    NA   NA  NA  NA  NA
SUM0:9     SUM0:15  0.000031    metal1     NA     NA    NA   NA  NA  NA  NA
SUM0:11    SUM0:14  0.000103    metal1     NA     NA    NA   NA  NA  NA  NA
SUM0:12    SUM0:14  0.000031    metal1     NA     NA    NA   NA  NA  NA  NA
SUM0:13    SUM0:14  0.000237    metal2     NA     NA    NA   NA  NA  NA  NA
SUM0:13    SUM0:16  0.000031    metal2     NA     NA    NA   NA  NA  NA  NA
SUM0:15    SUM0:17  0.000031    metal1     NA     NA    NA   NA  NA  NA  NA
```

Detailed Report

SYNOPSYS®

Tx-level: report_total_net_capacitance

Calculate and report the total capacitance of a net

Usage:

starrc_shell> report_total_net_capacitance \

<list or collection of nets>

For nets with escape character, use “get_nets –exact”

e.g. report_total_net_capacitance [get_nets -exact {net\\[0\\]}]

Output:

- List of nets provided with their total capacitance

```
starrc_shell> report_total_net_capacitance   "SUM0 B0"
========  ================
Net Name  Total Capacitance
========  ================
SUM0      0.013721
B0        0.089779
```

Output Report

SYNOPSYS®

Tx-level: report_routed_nets

Reports all nets on specified layer (e.g. top AP layer)

Usage:

starrc_shell> report_routed_nets \

-layer “list of layers”

Output:

- List of nets, total capacitance and percentage layer contribution to total capacitance

```
starrc_shell> report_routed_nets -layer "metal2 metal3"
=========================================
Total number of nets routed on metal2: 16
=========================================
Net Name  Total Capacitance  metal2 Capacitance  %metal2/Ct
========  =================  ==============      ======
A0        0.092668           0.012534            13.525705
A1        0.092722           0.012547            13.531848
A2        0.092721           0.012547            13.531994
A3        0.092718           0.012544            13.529196
B0        0.089779           0.012808            14.266142
B1        0.089778           0.012809            14.267415
B2        0.089778           0.012809            14.267415
B3        0.089776           0.012809            14.267733
CIN       0.064663           0.003958            6.120966
SUM0      0.013721           0.007130            51.964143
SUM1      0.013721           0.007130            51.964143
SUM2      0.013721           0.007130            51.964143
SUM3      0.013721           0.007130            51.964143
n6        0.066932           0.003958            5.913464
n7        0.066931           0.003958            5.913553
n8        0.066931           0.003958            5.913553
=========================================
Total number of nets routed on metal3: 0
=========================================
Net Name  Total Capacitance  metal3 Capacitance  %metal3/Ct
========  =================  ==============      ======
```

Output Report

SYNOPSYS®

Tx-level: report_length_layerwise

Reports the layerwise length distribution of a net

Usage:

starrc_shell> report_length_layerwise \

-of_objects “list or collection of nets”

Output:

- Layerwise length report of nets
- Requires NETLIST_TAIL_COMMENTS: YES during extraction

```
starrc_shell> report_length_layerwise -of "SUM0 B0"
ERROR: Length attribute does not exists. Please check if extraction was run with NETLIST_TAIL_COMMENTS: YES
```

```
starrc_shell> report_length_layerwise -of_objects “SUM0 B0”
==================================
Net: SUM0
Report: Layerwise length of net
==================================
metal1 21u
metal2 55.5u
==================================
Net: B0
Report: Layerwise length of net
==================================
metal1 252.5u
metal2 113u
poly   189u
```

Output Report

SYNOPSYS®

Tx-level: report_bounding_box

Reports the approximate bounding box of a net

Usage:

starrc_shell> report_bounding_box \

-of_objects “list or collection of nets”

Output:

- Net with llx, lly, urx, ury coordinates
- Coordinates of ground capacitor nodes are used for computation

```
starrc_shell> report_bounding_box  -of_objects “SUM0 B0”
======== ========== ========== ========== ==========
Net Name llx        lly        urx        ury
======== ========== ========== ========== ==========
SUM0     -467.000000 11.000000  -458.000000 82.000000
B0       -497.000000 2.500000   -272.000000 82.000000
```

Output Report

Tx-level: report_rc_contribution

Reports the approximate bounding box of a net

Usage:

starrc_shell> report_rc_contribution \

-of_objects “list or collection of nets”

Output:

- Report of net name, total capacitance, total resistance and layer wise % contribution for capacitance and resistance

```
starrc_shell> report_rc_contribution   -of "SUM0 B0"

================================
Net Name: SUM0
Total Resistance: 0.003476 KOhms
Total Capacitance: 0.013721 pF
Layer     ResValue(KOhms) %ResContribution CapValue(pF) %CapContribution
=====     =============== ================ ============ ================
SUBSTRATE 0.001100        31.645570        0.002219     16.172291
metal1    0.000449        12.917146        0.004372     31.863567
metal2    0.001927        55.437284        0.007130     51.964143

================================
Net Name: B0
Total Resistance: 0.581682 KOhms
Total Capacitance: 0.089779 pF
Layer  ResValue(KOhms) %ResContribution CapValue(pF) %CapContribution
=====  =============== ================ ============ ================
metal1 0.007996        1.374634         0.051307     57.148108
metal2 0.005152        0.885707         0.012808     14.266142
poly   0.568534        97.739658        0.025664     28.585749
```

Output Report

SYNOPSYS®

Tx-level: report_net_connectivity

*Report *P, *I and cells connected to a net*

Usage:

starrc_shell> report_net_connectivity \

<“list or collection of nets”>

For nets with escape character, use “get_nets -exact”

e.g. report_net_connectivity [get_nets -exact {net\\[0\\]}]

Output:

- Report with *P, *I and cells
- *P report contains pin name, direction and x/y coordinates
- *I report contains port name, direction, cell name, x/y coordinates
- Cell report contains name, bounding box x/y coordinates

```
starrc_shell> report_net_connectivity SUM0

============================
Net: SUM0
Report Type: Net Connectivity
============================
------- --------- ------------   ------------
*P Name Direction x-coordinate   y-coordinate
------- --------- ------------   ------------
SUM0    out       -459500.000000 81000.000000
-------   --------- ----    ------------   -----------
*I Name   Direction Cell    x-coordinate   y-coordinate
-------   --------- ----    ------------   -----------
0/33/M2/s inout     0/33/M2 -462500.000000 36250.000000
0/33/M1/s inout     0/33/M1 -462500.000000 11000.000000
----    ---------------- ---------------- ---------------- ----------------
Cell    x-coordinate min y-coordinate min x-coordinate max y-coordinate max
----    ---------------- ---------------- ---------------- ----------------
0/33/M2 -462500.000000   36250.000000     -462500.000000   36250.000000
0/33/M1 -462500.000000   11000.000000     -462500.000000   11000.000000
```

Detailed Connectivity Report

SYNOPSYS®

Tx-level: report_top100cg_corners

Reports top 100 nets with largest ratio of ground cap between parasitic corners

Usage:

starrc_shell> report_top100cg_corners \

-parasitic_corners <corner1 corner2>

Output:

- List of nets with ratio of Cg (corner1/corner2)
- The Cg for each net is summed up when computing Cg1/Cg2

```
============================
Report Type: Top 100 Cg Variation
Parasitic Corner 1: typ
Parasitic Corner 2: max
============================
Net                          Ratio
===                          ====
sec_msb/cnt_blk1/n181        1.054146
min_msb/cnt_blk1/n224        1.052174
min_msb_led[3]               1.051689
min_msb/conv_blk1/n122       1.050078
min_msb_led[4]               1.049475
min_lsb_led[0]               1.049014
sec_lsb/cnt_blk1/n145        1.048507
min_msb_led[5]               1.047774
sec_lsb/cnt_blk1/reg_in[1]   1.046881
...
```

Output

Tx-level: report_dominant_layer_in_path

Reports the R and C dominant layer for a specified path (PT) or list of nets (StarRC/PT)

Usage:

starrc_shell> report_dominant_layer_in_path \

[-from <start point>\]

[-to <end Point>\]

[-of_objects <list or collection of nets>]

Output:

- The options -from/-to are only supported in PT (uses get_timing_paths)
- The -nets option is supported in StarRC and PT

```
starrc_shell> report_dominant_layer_in_path  -of "SUM0 B0"

List of nets in specified timing path:
net 1: SUM0
net 2: B0
Total number of nets in the timing path: 2

R dominant layer: poly
Total R on poly: 0.568534

C dominant layer: metal1
Total C on metal1: 0.055679
```

Output

SYNOPSYS®

Tx-level: report_point_to_point_resistance

Reports P2P R for all pin/instance port combinations of the specified net

Usage:

starrc_shell> report_point_to_point_resistance \

-of_objects <list or collection of nets>

Output:

- Pin1, pin2 and equivalent P2P R
- All combinations of *P/*I are computed

```
starrc_shell> report_point_to_point_resistance -of_objects "SUM0 B0"
************************
Net: SUM0
Report Type: P2P R
************************
Pin1       Pin2       P2P R
====       ====       =====
SUM0       0/33/M2/s  0.002518
SUM0       0/33/M1/s  0.002802
0/33/M2/s  0/33/M1/s  0.001385
************************
Net: B0
Report Type: P2P R
************************
Pin1       Pin2       P2P R
====       ====       =====
B0         0/38/M5/g  0.051076
B0         0/38/M2/g  0.027876
B0         0/53/M3/g  0.059847
B0         0/53/M1/g  0.036113
B0         0/49/M3/g  0.046901
B0         0/49/M1/g  0.043435
B0         0/54/M5/g  0.043880
B0         0/54/M2/g  0.043079
0/38/M5/g  0/38/M2/g  0.074400
0/38/M5/g  0/53/M3/g  0.108194
0/38/M5/g  0/53/M1/g  0.084461
0/38/M5/g  0/49/M3/g  0.095249
```

Output

New Features in Q-2019.12 for Tx-Level Flow

- All Angle Extraction On By Default
- Standalone Reducer
- Mutual Inductance Extraction
- Transistor Level Performance Improvement
- Transistor Level GPD Support for Additional Options
- Exploding Trivial Instance Ports
- Parasitic Explorer – Tx Shell
- QuickCap Partition Speed-up

QuickCap Partition Speedup

- Quickcap is a 3D field solver. Using Quickcap to extract a lot of nets may be slow. In order to improve the runtime and efficiency to extract designs with a lot of nets, a speed-up is required.
- The objective is to support “-partition” Quickcap run. Then quickcap run can use a lot of lsf machines efficiently, where accuracy impact is small.
- A new option is available to define window size, margin for each window, and fringe to specify the boundary to ensure the integration surfaces are included.
- Quickcap run can use –LSF for independent partition runs
- A –totalC option can be used to optimize in a partition run. For example, with a setting “-totalC 30,8,2,1”, quickcap first runs with 30% goal and output totalC, then 8%, 2%, and final 1%. Each run is based on latest totalC results.

QuickCap Partition Speedup: User Interface

- Usage option for quickcap partition run:
  - "-partition <window> <margin> <fringe> <-totalC 20,5,2,1> –LSF <num>"
  - Default values:
    - <window> based on number of nodes and number of cores
      - window = sqrt((xSize+0.2um)*(ySize+0.2um)/numPartitions), cutoff 0.5um <= window <= 2um
      - numPartitions = totalNumNet / 200(avgNumNetPerPartition)
    - <margin> based on number of nodes
      - If netNum <= 30k, 5um
      - If 30k < netNum < 210k, internal algorithm is used to calculate <margin> based on memory/net relation
      - If netNum > 210k, 2um
    - <fringe> 0.1um
    - <-totalC> "20,5,2,1"
  - Before –LSF run, make sure to set LSF farm environment parameter, for example
    - setenv LSF_STR 'qsub -P bnormal -l "qsc=m mem_free=4.0G"'
  - Currently supports "-d" accuracy goal option only, otherwise fallback to non-partition runs.

QuickCap Partition Speedup: Special Situations

- Scenarios to turn off –partition with –lsf
  - Number of nodes < 4000
  - “-g pct”, “-g capacitance”, “-g pct@capacitance” “-g pct@capacitance” “-g pct1@value...pct2”

- Error and Warning
  - *.cap file should come from latest version gds2cap. Previous release’ *cap may not work for this function.
    - ERROR: quickcap cannot find “layoutMerics” in header of cap file while –partition is on.
  - If number of lsf is larger than number of partition
    - WARNING: When number of partitions is less than number of cores, reduce the number of cores to number of partitions and send out warning message
  - If number of nodes is less than 4k or “-g” is used
    - WARNING: Revert to flat run for number of nodes < 4000 or “-g” is used

New Features in Q-2019.12 Common for both flows

- Re-implementation RVWS(RPSQ_VS_WIDTH_AND_SPACING and RHO_VS_WIDTH_AND_SPACING)
- Compare Parasitics - Multi Process/DP
- Compare Parasitics - XY pin matching
- Different DOB Per Corner in SMC Flow
- Negating Wild Cards In Power Nets

Re-implementation RVWS(RPSQ_VS_WIDTH_AND_SPACING and RHO_VS_WIDTH_AND_SPACING)

	Current behavior	2019.12

	• grdgenxo converts RPSQ/RHO_VS_WIDTH_AND_SPACING to RES **EVWS** in nxtgrd
	• grdgenxo converts RPSQ_VS_WIDTH_AND_SPACING to RES EVWS and write RVWS in TLUPlus for IC Compiler II
	• StarRC uses RES **EVWS** for RES calculation
		• grdgenxo writes RPSQ/RHO_VS_WIDTH_AND_SPACING as RVWS table directly in nxtgrd
		• grdgenxo writes RPSQ/RHO_VS_WIDTH_AND_SPACING as RVWS table into TLUPlus directly
		• StarRC uses RVWS table directly during extraction in all scenarios
		  • Use drawn width and spacing to look up RVWS table for RES calculation
		  • When index is out of range during table lookup, use the border value

Issue	• Wrong $SI_width may be reported • RES may be mismatch vs. hand calculation • Confusion/accuracy for IC Compiler II	

Re-implementation RVWS(RPSQ_VS_WIDTH_AND_SPACING and RHO_VS_WIDTH_AND_SPACING)

- New behavior is only enabled by 2019.12 both StarRC and grdgenxo
- No new or changed commands needed, but need to regenerate nxtgrd with 2019.12

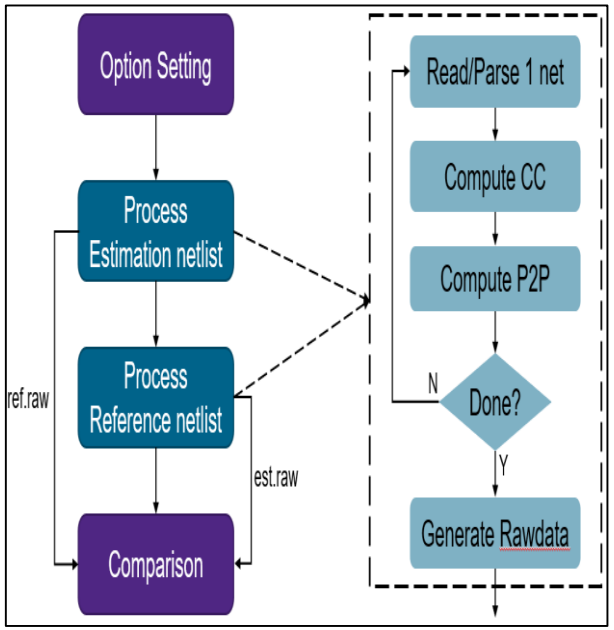
	Old NXTGRD file	2019.12 NXTGRD file
Old StarRC	Old behavior	If not crash, result might not be accurate.
2019.12 StarRC	Old behavior	New behavior

New Features in Q-2019.12 Common for both flows

- Re-implementation RVWS(RPSQ_VS_WIDTH_AND_SPACING and RHO_VS_WIDTH_AND_SPACING)
- Compare Parasitics - Multi Process/DP
- Compare Parasitics - XY pin matching
- Different DOB Per Corner in SMC Flow
- Negating Wild Cards In Power Nets

Compare Parasitics - Multi Process/DP

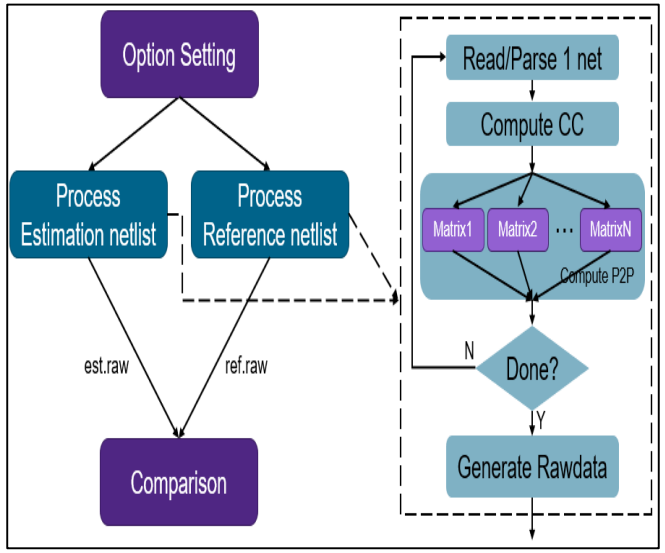
- Non-DP: Current compare parasitics processes estimation netlist and reference netlist in series.



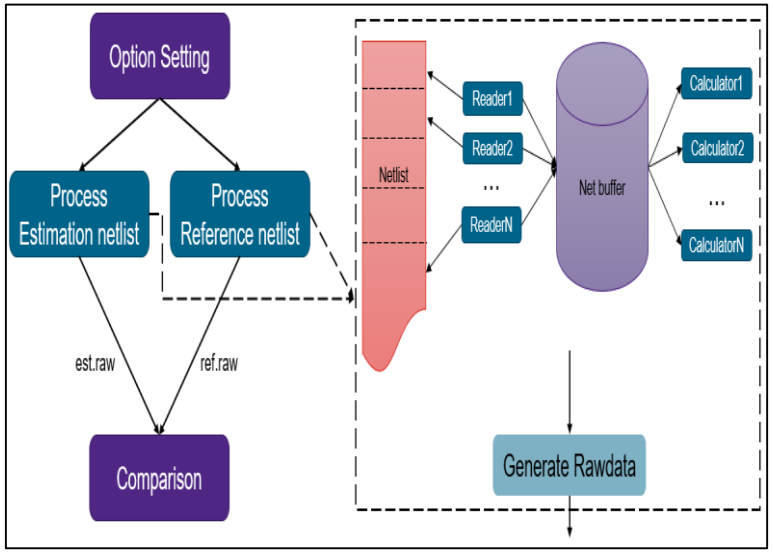
Flowchart of compare parasitics: non-DP

DP: The two netlist files are distributed into two different processes to run in parallel

- SPF/GPD Improvement
  - After processing of netlists, nets are handled sequentially
- SPEF Improvement
  - Handle nets in parallel. A consumer/producer-like paradigm is implemented



SPF/GPD



SPEF

Flowchart of compare parasitics: DP

SYNOPSYS®

User Interface for enabling DP compare_parasitics

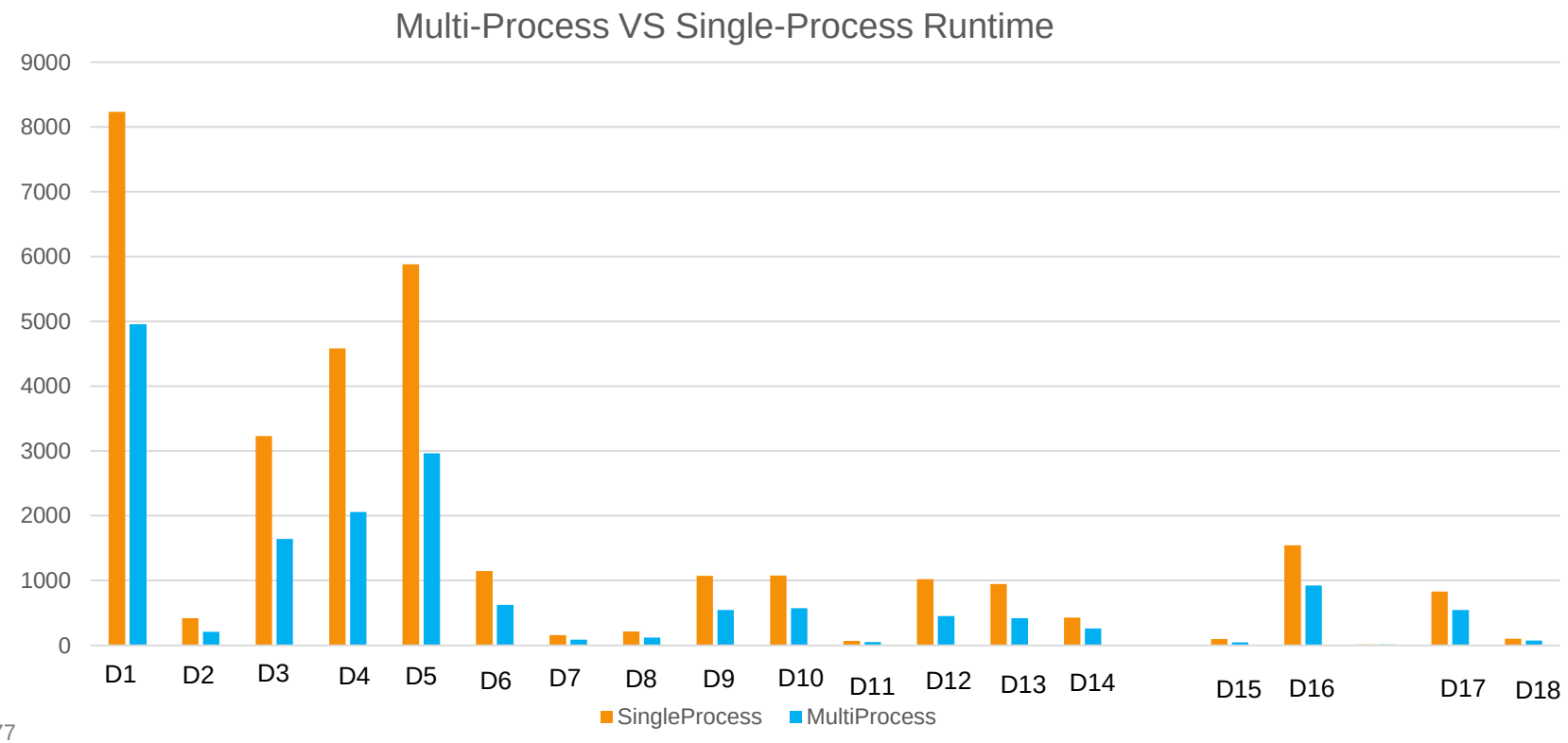
- New command line option

  -cores {number of cores} ← Specify number of core to run multi-process

  Usage:

```
$ StarXtract -compare_parasitics est.spef ref.spef -res 50 -cores 4
$ StarXtract -compare_parasitics est.gpd ref.gpd -corner "cbest_-40" -cores 5
```

Results for 4 cores, ~1.9X runtime improvement



SYNOPSYS®

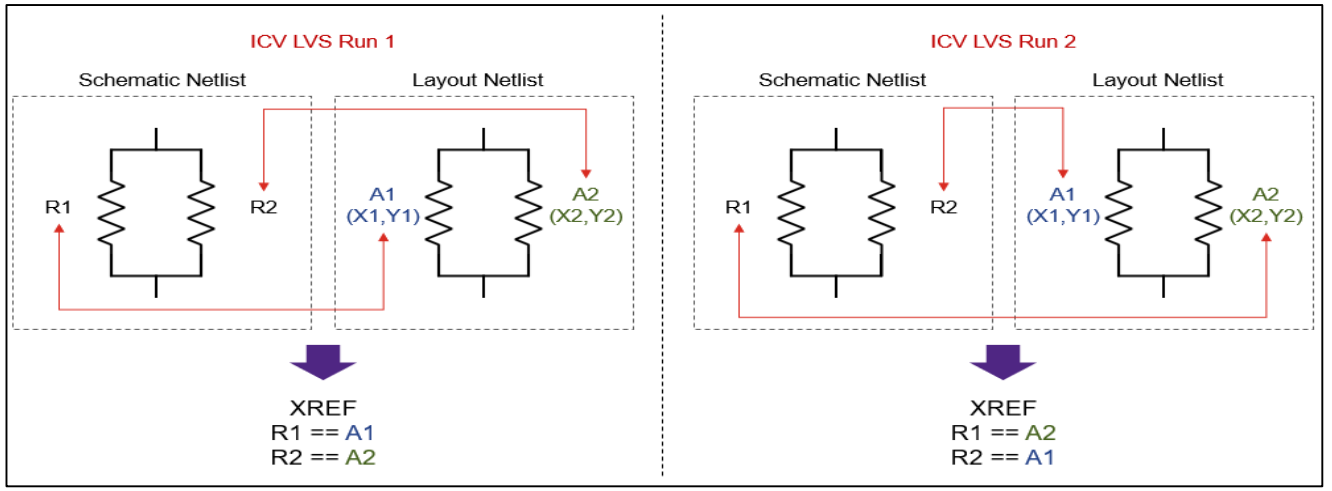
New Features in Q-2019.12 Common for both flows

- Re-implementation RVWS(RPSQ_VS_WIDTH_AND_SPACING and RHO_VS_WIDTH_AND_SPACING)
- Compare Parasitics - Multi Process/DP
- Compare Parasitics - XY pin matching
- Different DOB Per Corner in SMC Flow
- Negating Wild Cards In Power Nets

Compare Parasitics - XY pin matching

- StarRC -compare_parasitics shows significant P2P resistance and delay variances on the same physical design with DSPF netlist from
  - GDS → ICV→ StarRC → DSPF
  - OASIS→ ICV→ StarRC → DSPF
- Root cause:
  - ICV fails to produce consistent XREF for symmetric devices/instances
  - Current compare parasitics further INCORRECTLY matches the pair of start/end pins

```
------------------- P2P Distribution ---------
RES threshold: 50.000 Ohm
Min Error: -100.000%    Max Error: 816.251%
Mean Error: 1.968%      Standard dev: 28.783%
Number of matched nets: 19499
```



Two ICV LVS run produce inconsistent but valid XREF

SYNOPSYS®

Compare Parasitics - XY pin matching

- New Behavior will be able to match symmetric devices/instances based on (x, y) coordinates.
- Results in consistent parasitic analysis (resistance and delay)

- **New command line options**

  -match {name|**xy**}: Pin/Port match mechanism (default: name)

  Specify match as **xy** to enable xy-based compare parasitics. Specify it as **name** to revert to original name-based compare parasitics.

  Usage:

```
$ StarXtract -compare_parasitics <your_rc_file> <reference_rc_file> -match xy
```


User Configurable Pin Selection for –match xy

- Options used in command line:

Command Option	Description
-from_pin *<x,y>*	To specify xy coordinate of starting pin with format (<x>,<y>) when running xy-based compare parasitics. The name of pin is not applicable to xy-based compare parasitics
-to_pin *<x,y>*	To specify xy coordinate of ending pin with format (<x>,<y>) when running xy-based compare parasitics. The name of pin is not applicable to xy-based compare parasitics.
-net_config *<config file>*	To specify xy coordinate of pin section for config_file. The name of pin is not applicable to xy-based compare parasitics

- **Usage Examples**

$ StarXtract -compare_parasitics <your_rc_file> <reference_rc_file> -match xy **-from_pin** "431.6,558"

$ StarXtract -compare_parasitics <your_rc_file> <reference_rc_file> -match xy **-to_pin** “285.6,590.5“

$ StarXtract -compare_parasitics <your_rc_file> <reference_rc_file> -match xy **-net_config** <config_file>

*Configuration file for –net_config option:
FROM_PIN: 431.6,558
TO_PIN: 284.4,233.3
FROM_TO_PINS: 431.6,558 284.4,233.3

SYNOPSYS®

Output of P2P/delay from –match xy

- The xy coordinate of pin/port and their corresponding net/pin/port names are preserved in resistance and delay report for xy-based compare parasitics → P2P.rpt

```
est.spf ref.spf %diff   Netname Pin1    Pin2    Pin1_test       Pin1_reference  Pin2_test Pin2_reference
195.881 195.903 -0.011  Net1    (4.3680,6.3980) (3.7155,5.4720) Est1_Port Ref1_Port Est1_M:Gate Ref1_M:Gate
350.677 350.699 -0.006  Net2    (4.3965,6.2880) (3.7155,5.4720) Est2_Port Ref2_Port Est2_M:Gate Ref2_M:Gate
```

- P2P error distribution consistent with name based one

```
-------------------- P2P Distribution ----------
RES threshold: 50.000 Ohm
Min Error: -0.014%      Max Error: 0.170%
Mean Error: 0.000%      Standard dev: 0.003%
Number of matched nets: 29827
```

xy-based compare parasitics

```
-------------------- P2P Distribution ----------
RES threshold: 50.000 Ohm
Min Error: -100.000%    Max Error: 816.251%
Mean Error: 1.968%      Standard dev: 28.783%
Number of matched nets: 19499
```

name-based compare parasitics

Compare parasitics P2P testing result

- Design Specifications: Tx level
- All showed improved results from “–match xy”

Design Size	Process	Test Strategy	RES			
			Mean	S.D.	Min	Max
2132 nets	7nm	names	-0.19%	4.37%	-100.00%	44.72%
		xy	0.00%	0.00%	-0.02%	0.02%
447 nets	14nm	names	-1.85%	13.54%	-100.00%	3.69%
		xy	0.00%	0.01%	-0.01%	0.06%
4376 nets	14nm	names	2.79%	76.61%	-100.00%	1735.77%
		xy	0.00%	0.02%	-0.18%	1.32%

Compare parasitics P2P testing result

- Design Specifications: Gate level
- All showed improved results from “–match xy”

Design Size	Process	Test Strategy	RES			
			Mean	S.D.	Min	Max
1355399 nets	7nm	names	-2.34%	5.86%	-96.47%	100.92%
		xy	0.02%	0.09%	-0.95%	1.61%
77135 nets	5nm	names	-0.19%	0.89%	-26.13%	1.09%
		xy	0.01%	0.03%	-0.29%	1.09%
104552 nets	7nm	names	-1.02%	1.53%	-31.02%	3.55%
		xy	0.02%	0.21%	-10.37%	2.72%

New Features in Q-2019.12 Common for both flows

- Re-implementation RVWS(RPSQ_VS_WIDTH_AND_SPACING and RHO_VS_WIDTH_AND_SPACING)
- Compare Parasitics - Multi Process/DP
- Compare Parasitics - XY pin matching
- Different DOB Per Corner in SMC Flow
- Negating Wild Cards In Power Nets

Different DOB Per SMC Corner

- Currently DENSITY_OUTSIDE_BLOCK (DOB) setting is a global parameter for all layers and all corners. But density outside block may change for different corner condition and would like StarRC to extend the support for corner based setting.

- Currently StarRC supports DENSITY_OUTSIDE_BLOCK setting for all layers and all corners.

- From 2019.12, StarRC supports corner based DENSITY_OUTSIDE_BLOCK. The new SMC corner file adds **DENSITY_OUTSIDE_BLOCK** option, such as

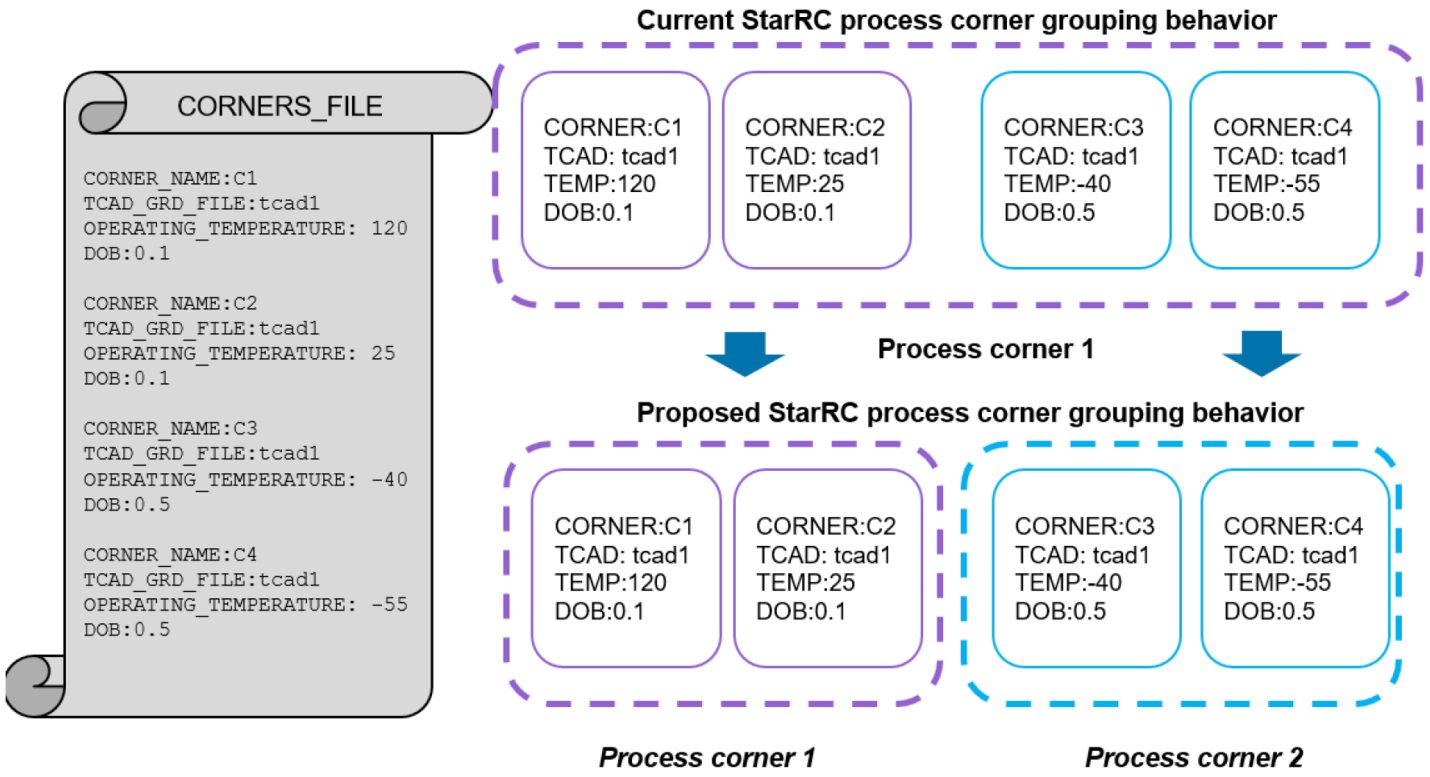
CORNER_NAME: ***name_of_corner***
TCAD_GRD_FILE: ***nxtgrd_path_and_file_name***
OPERATING_TEMPERATURE: ***temperature_in_Celsius***
(optional) CORNER_TYPE: **NOMINAL | OTHER**
(optional) MAPPING_FILE: ***map_file_name***
(optional) VIA_COVERAGE_OPTION_FILE: ***via_file***
(optional) **DENSITY_OUTSIDE_BLOCK**: ***floating number***

* Floating number is between 0.0 and 1.0

SYNOPSYS®

Different DOB Per SMC Corner

Introducing DOB to SMC corner impacts number of process corners as the process corner will be combination of unique nxtgrd and unique DOB value. For example, single nxtgrd and two different DOBs setting will extend process corner from original one process corner to two process corners.



SYNOPSYS®

New Features in Q-2019.12 Common for both flows

- Re-implementation RVWS(RPSQ_VS_WIDTH_AND_SPACING and RHO_VS_WIDTH_AND_SPACING)
- Compare Parasitics - Multi Process/DP
- Compare Parasitics - XY pin matching
- Different DOB Per Corner in SMC Flow
- Negating Wild Cards In Power Nets

Ability to Negate POWER_NETS Wildcards

- POWER_NETS
  - defines which nets are power nets.
  - only support select all wildcard(*) or partial wildcard(vss*).
- The negation wildcard(!) is not supported in current StarRC version.
- If a negation wildcard is used in POWER_NETS, the negation will not work, and a warning message SX-0722 (WARNING: Negation wildcard is not supported for power nets selection. Token ‘xxx’ will be ignored.)
- In 2019.12, StarRC will skip the POWER_NETS input start with negation(!).
- Other POWER_NETS will be processed as expected.

Ability to Negate POWER_NETS Wildcards

- User Interface

  POWER_NETS: <net names>

- Net names can be:
  - –Normal net name
  - –Partial wildcard (example: vss*)
  - –Select all wildcard (*)
  - –Negation wildcard(New supported, example: !vss, !vss*, !*)

SYNOPSYS®
Silicon to Software™

Thank You