Navigating the Labyrinth: Unravelling Challenges in Analog Circuit Debugging with Synopsys StarRC Parasitic Explorer

Venkat Nachiappan
Intel Corporation

Senthil Annamalai
Synopsys Inc

Introduction

Introduction

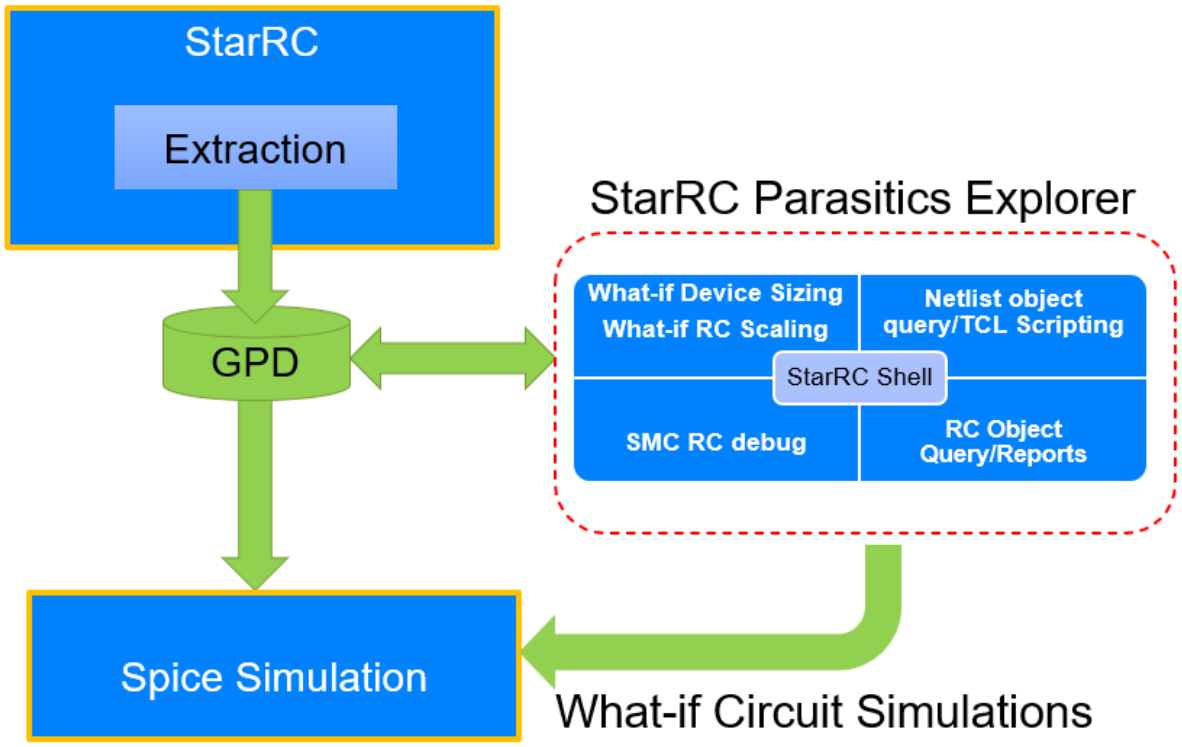
- Traditional methods existing today to assess impact of interconnect resistance and capacitance or examining effects of device sizing and drive strength changes on Analog circuit performance involves a costly design loop.
- This encompasses generating several versions of layout followed by extracting and capturing the parasitics into a Detailed Standard Parasitic Exchange (DSPF) netlist for simulation with an Analog Spice Circuit Simulation Engine downstream. Or other methods involve running simulation iterations in pre-extracted mode that don't capture the design intent accurately or manual intervention of capturing intent directly with ASCII RC netlists which are not efficient and scalable.
- Cover overview for an interconnect scaling and device parameter sizing flow for custom transistor level extraction context aiding designers to perform what-if RC analysis
- The framework aids designers with several different applications like simulating effects of variation in device drive strengths as PDK matures or working through post layout simulation debug prior to committing physical layout changes
- The flow additionally also provides a method to scale interconnect resistance and capacitive effects seamlessly to mimic and simulate effects of Process Design Kit back-end metal RC changes.

Where Does “what-if” Fit in the Design Flow

Design Exploration/Tech Readiness	Lead foundational IP’s on a process like PLL trying to catch up with Process Design Kit updates early on or teams working with PDK development and assess technology impact.
Design Impact Assessment	Review impact from changes in backend metal interconnect RC from current process kit revision to next on mature designs.
Design Analysis/Debug	Post Layout Simulation Debug as part of design process to detect and fix design specification or Functionality Simulate metal RC effects before initiating layout changes Simulate effects of device drive strength changes from one process design kit to another prior to locking on floorplan.

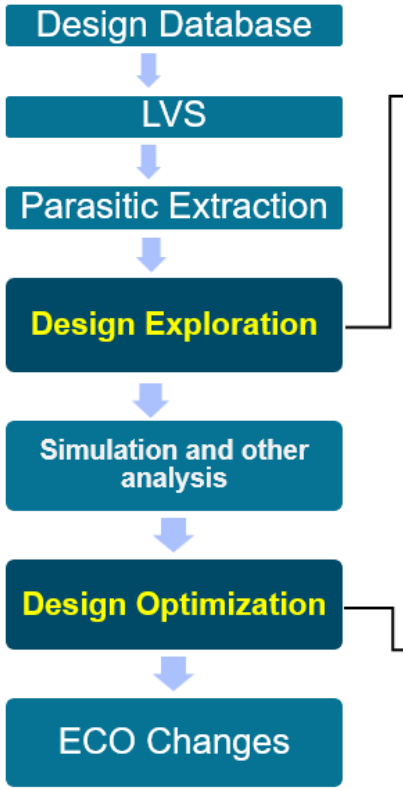
Parasitic Explorer Flow

Parasitic Explorer Flow



- Synopsys StarRC™ Parasitic Explorer Flow enables end users to perform various RC debug experiments from a single cockpit in addition to extending the capabilities for performing what-if type simulation experiments
- The primary input source for the flow is a Golden Parasitic Database generated by Synopsys StarRC with specific attributes for giving end users capability to perform tne needed RC debug and what-if simulation tasks
- StarRC PE has several extensive capabilities but in this presentation, we will focus on the Interconnect Scaling and Device Parameter sizing aspects feeding downstream spice simulation
- In the context of the Device Sizing and Interconnect Scaling aspects, end users start with an extracted database and iterate over that for the design exploration.
- Physical layout at some level of maturity in the design cycle with LVS'able where end user is able to run Layout versus Schematic Checks followed by extraction and then continue with the analysis depending on what context it is (Tech Readiness, Design Debug, etc).

Use Models for Design Debug and Exploration



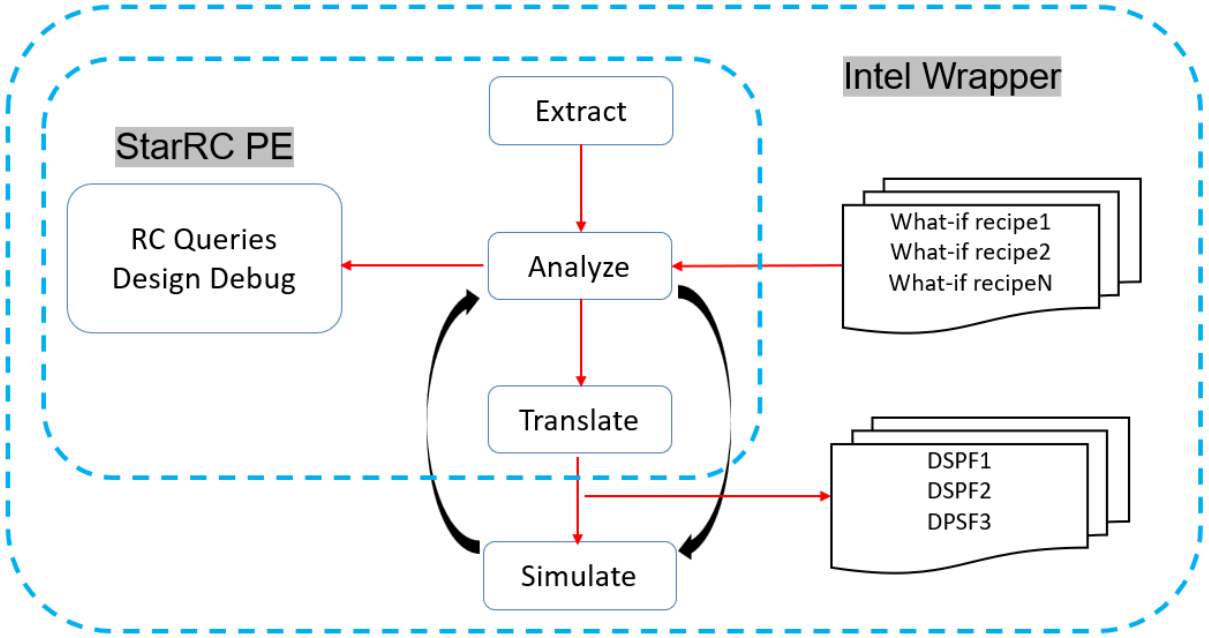
Gain deeper understanding of design before running expensive post layout simulations

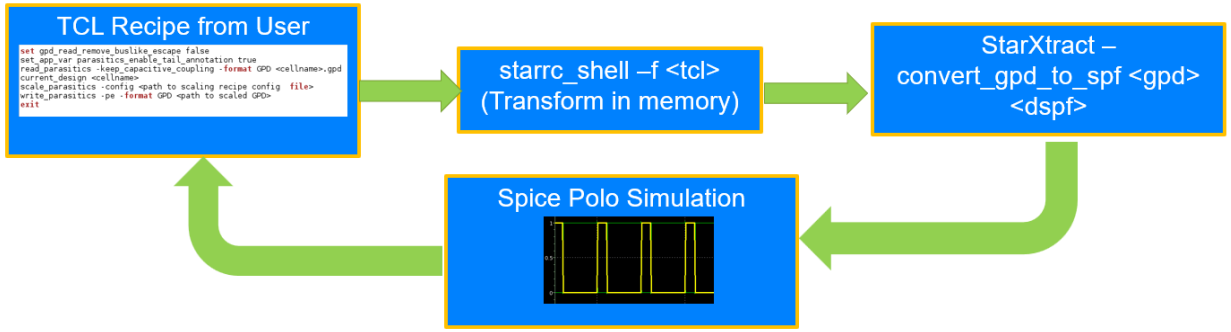
- What is the highest fanout ?
- Do I have devices with appropriate drive strength?
- What is the Elmore delay of my nets?
- Can I understand the Xtalk effect on critical signal nets?

Probe into specific target nets and its drivers receivers to uncover design issues

- Why is my design not meeting functional spec and PPA targets?
- Can I isolate the bottlenecks in the layout layers causing high R, higher coupling regions,etc?
- How is process migration/technology porting going to impact design functionality and margins?

Intel What-if Simulation Workflow





- Flow chart shows the Intel wrapper flow built around Synopsys StarRC Parasitic Explorer.
- The wrapper flow takes in recipes from end user and uses Synopsys StarRC parasitic explorer to transform changes in memory followed by writing out a Detailed Standard Parasitic Format (DSPF) RC netlist for Circuit simulations.
- The flow works to generate scaled DSPF at single temperature and interconnect skew or with StarRC simultaneous multi corner (SMC) mode to generate DSPF's across multiple temperature and interconnect skews.

Config File Syntax

1. Net Based Controls
   a) -net_list <net name> -res_factor <res_factor> -cc_factor <cc_factor> -gc_factor <gc_factor> //specify factor to “1” if no scaling is required
   b) -net_list <net name> -res_factor <res_factor2> -cc_factor <cc_factor2> -gc_factor <gc_factor2>
   c) -net_list <net name> -from <pin/port/node name> -to <pin/port/node name> -res_factor <res_factor> -cc_factor <cc_factor> -gc_factor <gc_factor>
2. Layer Based Controls
   a) -layer <layer name> -res_factor <res_factor> -cc_factor <cc_factor> -gc_factor <gc_factor> //specify factor to “1” if no scaling is required
   b) -net_list <net name> -layer <layer name> -res_factor <res_factor> -cc_factor <cc_factor> -gc_factor <gc_factor> //specify factor to “1” if no scaling is required
3. Width Based Controls
   a) -layer <layer name> -target_width <width> -res_factor 10
4. Device Sizing Controls
   a) device instance_name -w new_w_data -nf new_nf_data -m new_m_data –l new_l
   b) device instance_name -model new_model_name –cellw new_cellw
5. SMC controls

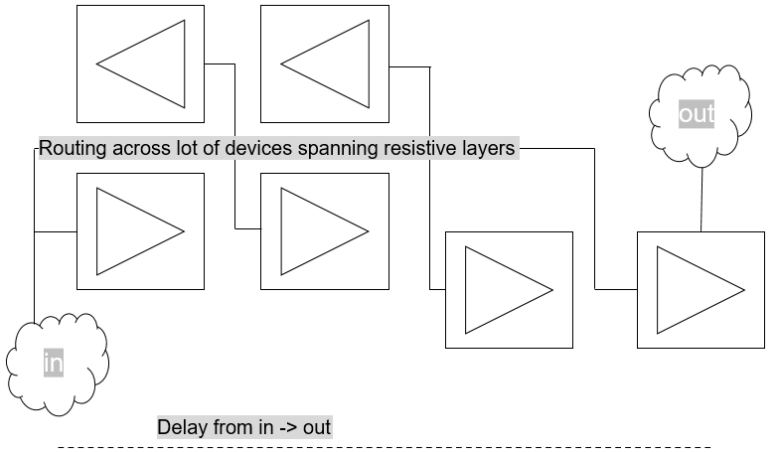
```
corner_start <temp1_skew1>
-net_list * -layer <layer name> -res_factor 2 -gc_factor 2 -cc_factor 2
corner_end

corner_start <temp2_skew2>
-net_list * -layer <layer name> -res_factor 2 -gc_factor 2 -cc_factor 2
corner_end

corner_start <temp3_skew3>
-net_list * -layer <layer name> -res_factor 10 -gc_factor 10 -cc_factor 10
corner_end
```

Design Usage Cases

Design Case 1



Design Scenario Details

- Case 1 demonstrates Design Exploration examining simulation results from emulating effects of different variants of layout routing topology for a level shifter before locking on floor plan.
- Sweep with different RC scaling recipes on critical nets to assess simulation impact and meet delay specifications across 4 simulation Corners2
- Critical net “NET1” is used for the evaluations with goal of meeting target delay of 1ns. Scope of analysis is to determine if promoting the routing to higher metal layers or widening the metal routing on the lower layers can meet the delay target
- 2 recipes run to show the upper and lower boundaries of the targeted delay margins. Experiments span reducing the overall RC by half (0.5x) in Recipe#1 and reducing the overall RC by 2x in Recipe#2

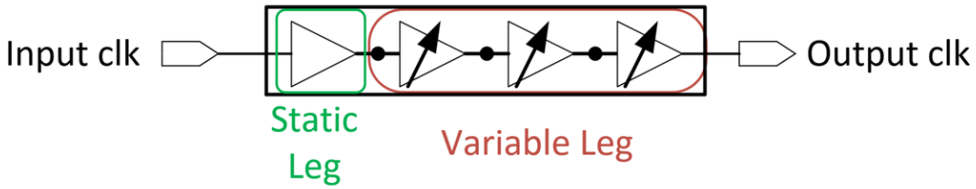
Scaling Recipe

Net	RC Profile	RC Scaling Recipe
Critical NET1	Vary MetalX by 2x and VIAX by 0.5x	-net_list NET1 -layer METALX -res_factor 2 -net_list NET1 -layer VIAX -res_factor 2
Critical NET1	Vary MetalX by 0.5x and VIAX by 2x	-net_list NET1 -layer METALX -res_factor 0.5 -net_list NET1 -layer VIAX -res_factor 0.5

Results

Recipe	Type	Spec	Pass/Fail	SIM_PVT1	SIM_PVT2	SIM_PVT3	SIM_PVT4
Recipe#1	Delay_Rise	<1ns	Fail	615.2ps	774.3ps	711p	1.205n
Recipe#1	Delay_Fall	<1ns	Fail	778.4ps	1.321ps	836.2p	891.1p
Recipe#2	Delay_Rise	<1ns	Pass	451.1p	563p	715.3p	512p
Recipe#2	Delay_Fall	<1ns	Pass	581p	674.1p	891.1p	609.2p

Design Case 2



Design Scenario Details

- Case 2 details scenario of Delay Linearization circuit analysis activity to determine optimum ratio of device sizes for achieving linear transitions across codes.
- Sweep device size ratio between groups of devices in the design.

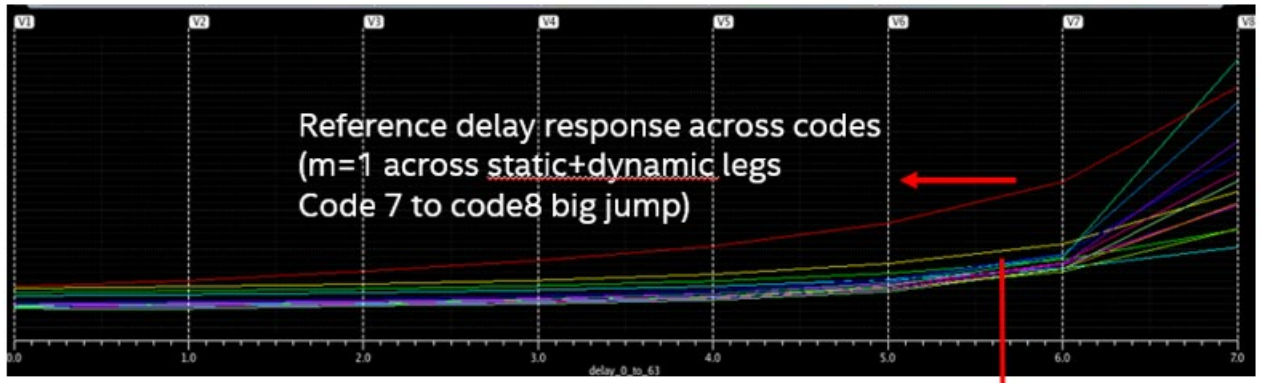
Design Scenario

- Static legs always on, dynamic legs controlled through code sweep.
- As codes sweep from 0 to 7 less of the dynamic TX legs are turned on.
- At code 7 only static legs on. Highest delay with no dynamic legs turned on.
- At code 0 all legs (dynamic+static) are on with fastest propagation (smallest delay) device sizes swept across static/dynamic legs and find sweep spot to make delay curve across codes linear with good range.

Dynamic	Static	Ratio of Dynamic to static	Device Sizing Recipe
1x	2x	1:2	-device <dynamic> -m 1 -device <static> -m 2
1x	3x	1:3	-device <dynamic> -m 1 -device <static> -m 2
1x	4x	1:4	-device <dynamic> -m 1 -device <static> -m 4
2x	1x	2:1	-device <dynamic> -m 2 -device <static> -m 1

Scaling Recipe

Design Case 2: Results

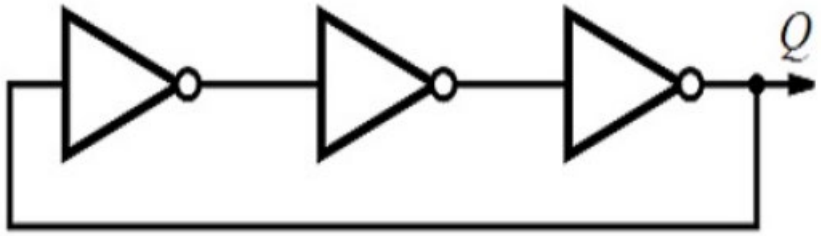


	V1	V2	V3	V4	V5	V6	V7	V8
	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0
	4.08526p	4.39093p	4.86975p	5.41678p	6.14986p	7.31863p	9.43896p	14.2769p
cd1x_s2x	3.98982p	4.0951p	4.21381p	4.40174p	4.71146p	5.25955p	6.25992p	8.94279p
cd1x_s3x	3.777p	3.87035p	3.96843p	4.12971p	4.36464p	4.75897p	5.4567p	6.99843p
cd1x_s4x	3.61161p	3.6792p	3.77025p	3.91086p	4.10459p	4.43684p	5.02546p	6.09862p
cd2x_s2x	3.26453p	3.32671p	3.44395p	3.6185p	3.90655p	4.47129p	5.73269p	10.8434p
cd2x_s3x	3.17631p	3.25147p	3.33544p	3.48299p	3.72823p	4.20846p	5.22479p	8.253p
cd2x_s4x	3.12413p	3.1541p	3.26039p	3.38051p	3.58819p	4.00188p	4.85617p	7.03292p
cd3x_s2x	3.1295p	3.20459p	3.27423p	3.44219p	3.71628p	4.30178p	5.71339p	13.4947p
cd3x_s3x	3.0343p	3.15812p	3.20717p	3.36941p	3.59337p	4.09375p	5.29929p	9.96436p
cd3x_s4x	2.98772p	3.10521p	3.18649p	3.27977p	3.50671p	3.95302p	5.00061p	8.35154p
cd4x_s2x	3.04255p	3.09759p	3.19179p	3.31198p	3.55369p	4.12603p	5.57919p	15.6867p
cd4x_s3x	3.00038p	3.06185p	3.15515p	3.27872p	3.47208p	3.94916p	5.22856p	11.5168p
cd4x_s4x	2.96477p	2.96005p	3.11308p	3.21166p	3.40348p	3.83421p	5.00083p	9.47139p

Sweep Spot for good range and
Linear response between codes
Static Legs = 2x,Dynamic Legs = 1x
Jump from 6'th to 7'th code
8.942279-6.2599=~2.6ps

Results

Design Case 3



Design Scenario

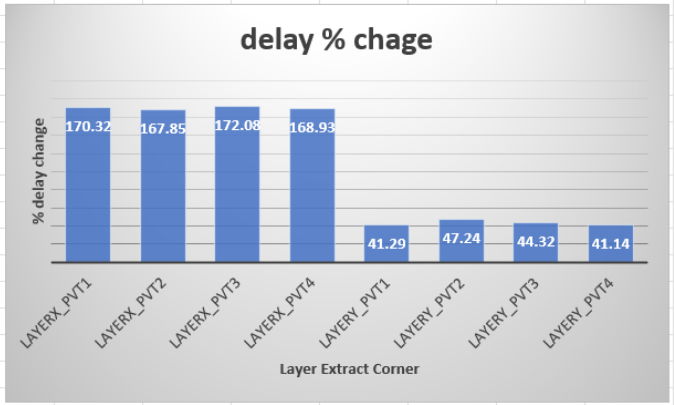
Net	RC Profile	RC Scaling Recipe
All nets	Scale MetalX	-net_list * -layer METALX -res_factor 2 corner_start <125C_tttt> -net_list * -layer METALX -res_factor 1.5 corner_end corner_start <100C_prcs> -net_list * -layer METALX -res_factor 0.5 corner_end
NA	PMOS:NMOS size ratio	-device NMOS1 -m 2 -device PMOS1 -m 4

Scaling Recipe

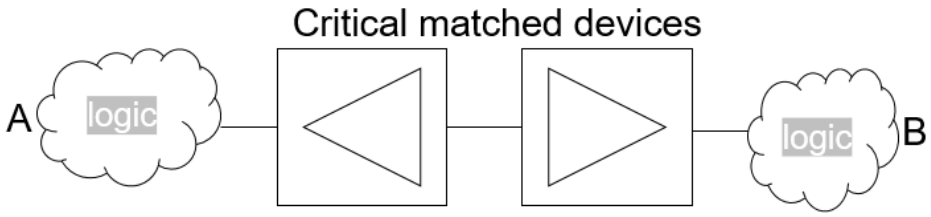
- Case 3 examines Tech Readiness Sensitivity Analysis Activity during early process assessment phase. We vary RC to detect which layers are more sensitive to delays across interconnect skews.
- GPD contains 4 sets of corners and when scaled and transformed through parasitic explorer will produce 4 scaled DSPF's for downstream spice simulations.
  starrc_shell> get_gpd_corners -gpd ringosc.gpd 125C_tttt 100C_tttt 125C_prcs 100C_prcs
- Simulation test bench device skews and voltages levels kept same sweeping across the 4 scaled DSPF's. Use model also aids to determine sweep spot for N to P MOS device sizing ratios meeting delay specifications for target frequencies.
- If the PMOS is weaker the sizing ratio sweep can provide guidance for end users to make design choices for meeting target delays at different frequencies.

Results

Extract Corner	delay % chage
LAYERX_PVT1	170.32
LAYERX_PVT2	167.85
LAYERX_PVT3	172.08
LAYERX_PVT4	168.93
LAYERY_PVT1	41.29
LAYERY_PVT2	47.24
LAYERY_PVT3	44.32
LAYERY_PVT4	41.14



Design Case 4



Design Scenario

Design Intent Recipe	Device Sizing Recipe	Delay
Recipe#1 Scale MXI1 and MXI2 devices Mult by from 4 to 8	-device XIBUF.MXI1* -m 2 -model ULVT -device XIBUF.MXI2* -m 2 -model ULVT	Original Delay – 356ps Scaled Delay – 345ps
Recipe#2 Scale MXI1 and MXI2 devices Mult by from 4 to 9	-device XIBUF.MXI1 -m 3 -model ULVT -device XIBUF/MXI2_NET*-m 2 -model ULVT	Original Delay – 356ps Scaled Delay – 337ps (closer to spec)

Recipe and Results

- Case 4 design Scenario covers observation of circuit propagation delays as a function of upcoming PDK device model changes and device drive strength variations.
- Delay spec to be met is 340pico seconds between input A and output B shown in the circuit snapshot with critical matched devices are in the middle of the design.
- Input A traverses through the critical matched devices playing a crucial role in controlling delay to output B.
- Device sizing recipe dialed in with the hierarchical device names and multiplier scaling factors.
- Results table explains the recipes.Recipe#1 scales the multiplier on MOS devices from 4 to 8 mimicking the effect of increasing the drive strength 2x. Scaled Delay obtained amounts to 345ps. Recipe# 2 bumps up the multiplier from 4 to 9 obtaining delay of 337ps and closer to meeting the target spec of 340ps.

PE Interactive Shell/Virtuoso Integration

Interactive Parasitic Design Analysis and Exploration

Synopsys StarRC Parasitic Explorer

- Dominant layer in a timing path
- 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 Analysis

Net Analysis

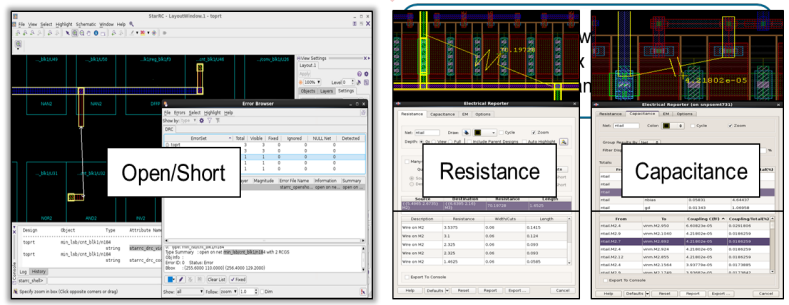
- Per-layer length distribution
- Nets routed on top metal layers
- Total & ground capacitance, resistance
- Width/layer of net segments
- Point-to-point equivalent resistance
- Shortest resistive path
- Non-physical resistors (for EM)

- Capacitance variation between corners
- Resistance variation between corners

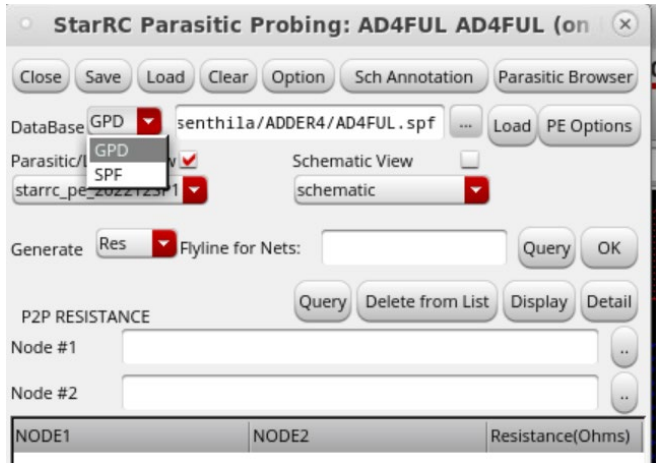
Corner Analysis

Design Debug

- Annotate design with RC topology
- Get bounding box of a net
- Traverse opens and shorts

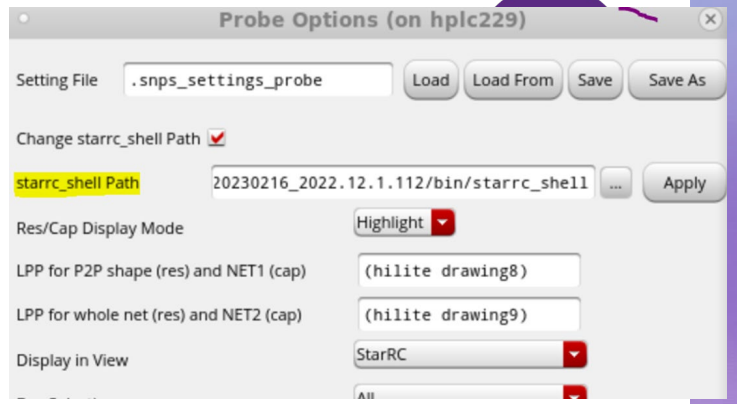


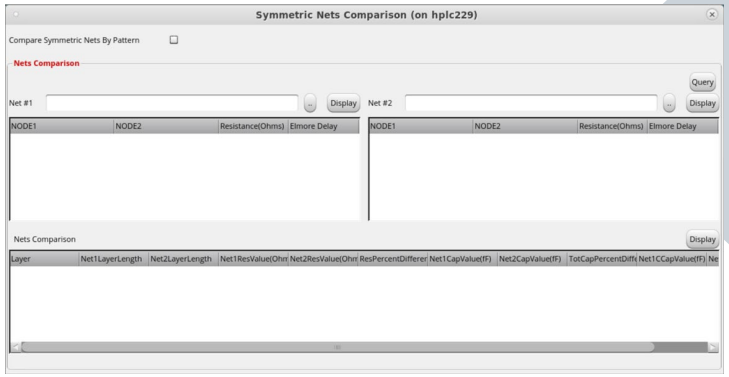
Parasitic Explorer – Virtuoso Interface



SPF Support

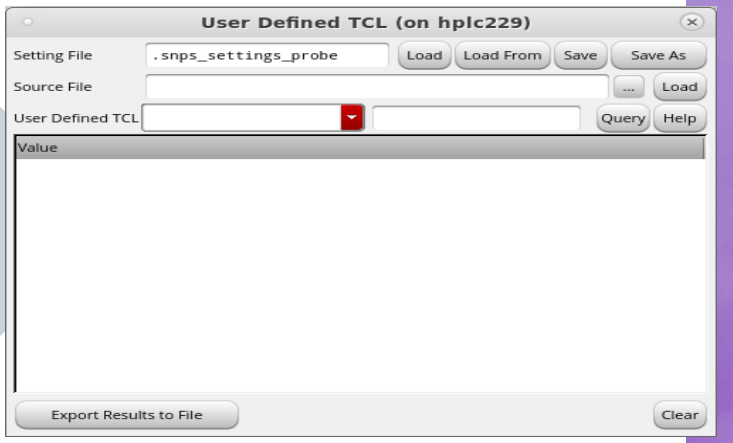
Version Independent

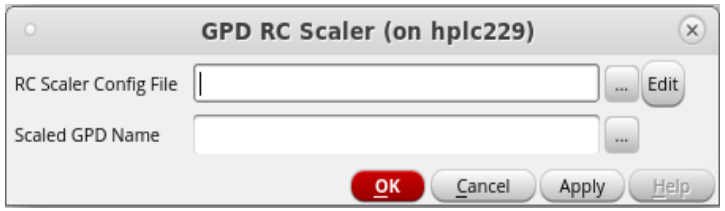




Symmetric net analysis

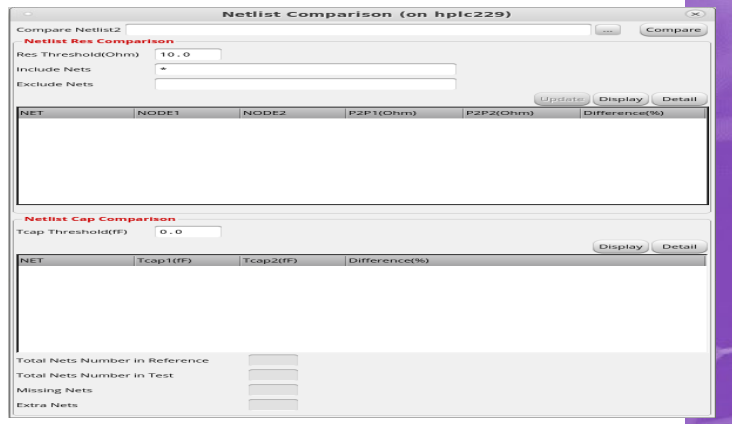
User defined TCL support





RC Scaler

PDK Comparison

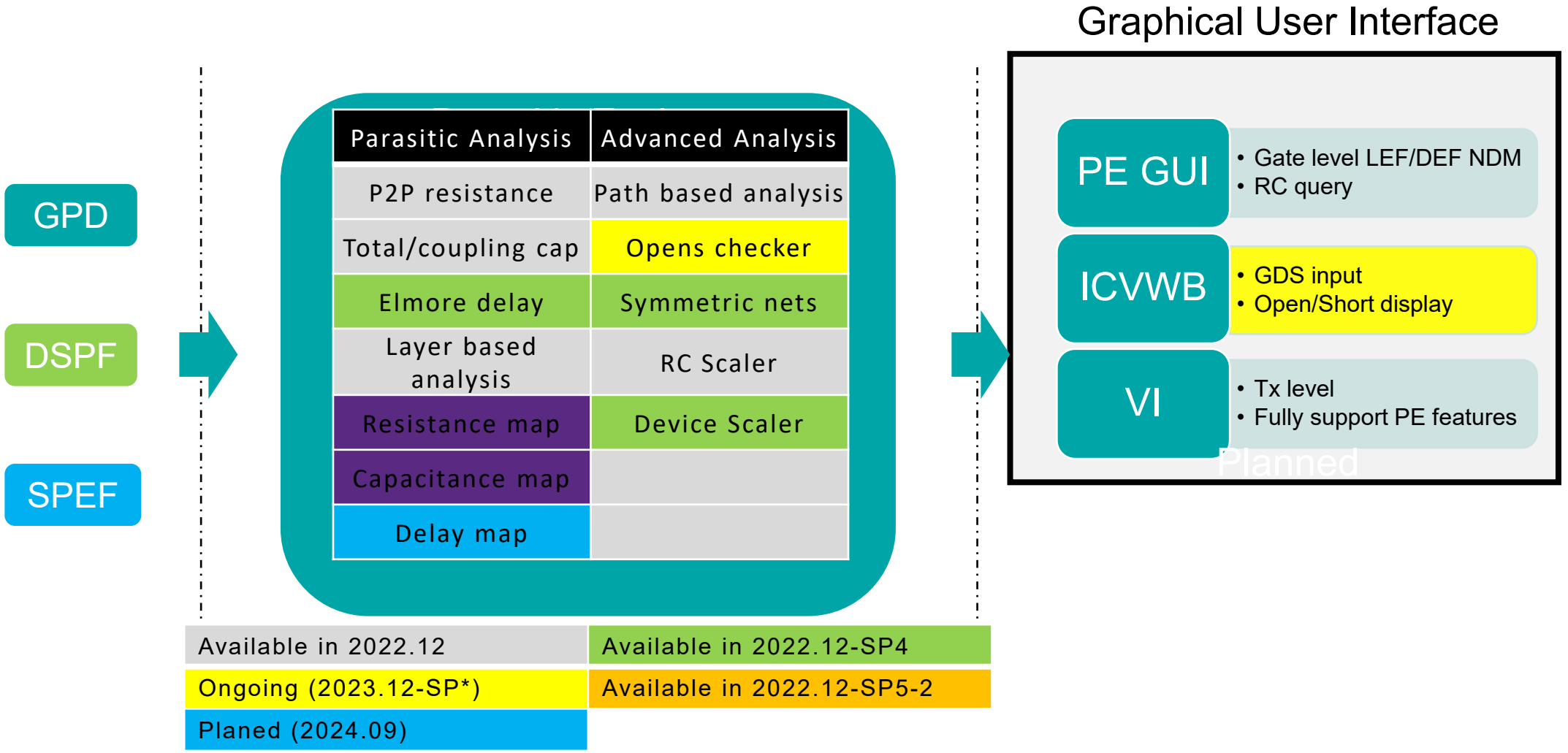


Conclusion and Future Plans

Conclusion and Future Plans

- In this presentation we have demonstrated a new novel capability where-in end users can perform circuit analysis on early or mature designs with shorter turn around times before making progress on design choices.
- The flow described uses the Synopsys StarRC infrastructure and wraps around custom Intel flow creating a user-friendly TCL driven framework. The capability aids analog mixed signal circuit designers to sweep device sizing across PVT's or sweep RC across different PVT's and quickly run circuit simulations to assess impact.
- Future Plans include additional device sizing controls and extend capability to working off a standalone DSPF RC netlist.

Parasitic Explorer Roadmap




THANK YOU
YOUR
INNOVATION
YOUR
COMMUNITY