

RTL Architect: Parallel RTL Exploration with Unparalleled Accuracy

Simply Better RTL

Author

Jim Schultz

Sr. Product Marketing
Manager, Synopsys

Shift-Left Revolution

Synopsys opened new doors to design creativity when it introduced RTL Architect, the industry's first physical aware, RTL analysis, exploration, and optimization environment. The tool's lofty goal was to leverage implementation and signoff engines to address power, performance, and area (PPA) targets at the earliest stages of the digital design cycle. Increasing chip complexity and restrictive advanced node rules have made it harder for implementation tools to achieve PPA targets and node entitlements via last-mile optimizations. RTL Architect enables designers to "shift-left" and predict the implementation impact of their RTL, as seen in Figure 1.

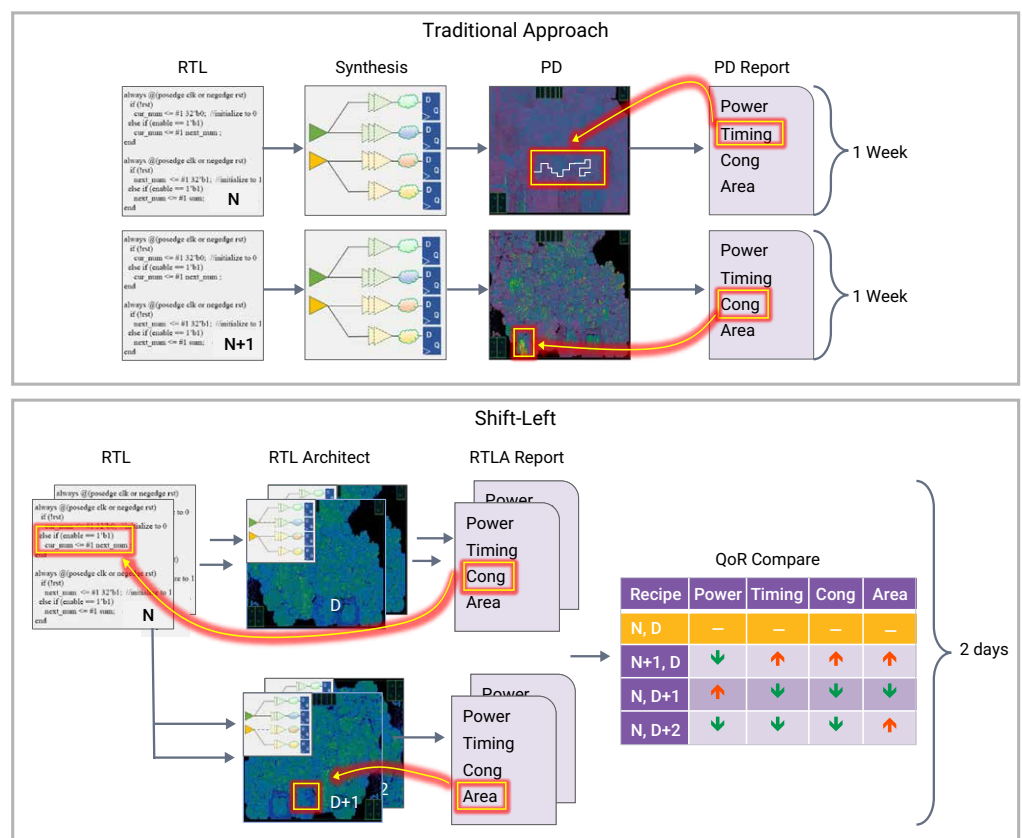


Figure 1: Traditional vs. Shift-Left Approach to RTL Implementation Feedback

RTL designers, SoC integrators, and IP developers have embraced this fast, predictive technology to give them new insight into their designs. They have been trail-blazing innovative ways to get better PPA, faster turn-around-time, and higher accuracy for design/IP implementation hand-off.

Experiment Explosions

One challenge is how to explore different design parameters in a systematic and efficient way. In traditional RTL design, a change is made, and the signal waveforms are reviewed to verify the desired functionality. Very little effort is made to optimize the RTL for PPA since it first must be synthesized and then physically implemented. Finally, the RTL designer must cross-reference the physical design (PD) timing, congestion, and power reports back to their RTL. This process overhead and lack of quality PD feedback has made design-for-PPA unattractive at the RTL stage, even though it has big potential benefits. The result is only half the team digging for PPA gold.

An RTL IP designer described this frustrating process. They know their logic well and how to optimize it for performance. However, they don't know where to look. Some PD teams that incorporate their IP may take a couple of weeks to get back to them, due to their busy schedules. When they do give feedback, it is often too late to make changes since the changes would impact the verification flow. Therefore, they want a way to confidently predict PPA issues themselves and a systemic approach to tune it before they hand-off to PD.

Since RTL Architect addresses both the flow overhead and quality feedback concerns, it is now easier for the RTL designer to explore design-for-PPA options which enables the entire team to dig for PPA gold. Setting up, tracking, and analyzing various experiments can be time-consuming and slow, but the tool's unique features offer a wide range of parameters to explore and a simple way to compare the results. In the following sections, we will examine how RTL Architect helps tackle this problem and describe some of the most common experiments it performs.

Parallel RTL Exploration (PRE)

RTL Architect provides an efficient infrastructure to setup, launch, and compare multiple design experiments. See figure 2.

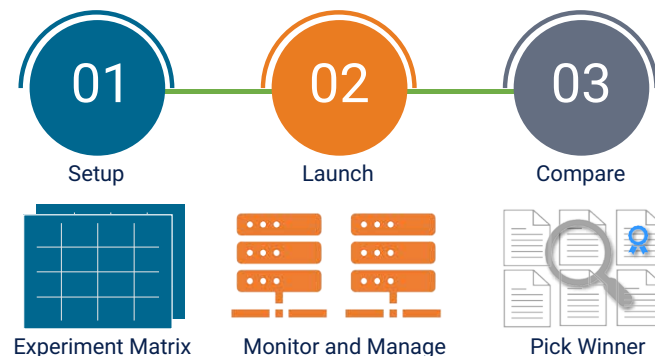


Figure 2: Parallel RTL Exploration (PRE) Steps

Setup

The setup feature allows the user to quickly define a large experiment matrix by defining sweep parameters and their corresponding range of values. For example, sweeping utilization over a range of {0.5, 0.7, 0.8} and sweeping aspect ratio over a range of {1:2 1:1 2:1}, will generate nine experiments. Additional sweep parameters will have a multiplicative effect on the number of jobs. An exclude option can remove unwanted experiment combinations. In addition, setup manages the distributed processing environment, including hosts, max processes, and max cores.

Prior to launching the experiments, both a summary and a detailed report can be generated to provide confidence in the experiments being run to ensure compute resources are not being wasted on unwanted runs.

Launch

Runs are automatically named based on their sweep parameters and saved to corresponding, separate, directories which makes the experiments easier to manage and compare. Job monitoring tracks the status of each run and the system resources used. It uses the robust job submission platform common to all tools on the Fusion Design Platform. Therefore, it honors host precedence, resource specifications, and even re-submission on failure which is important if a job dies due to machine memory limitation.

Compare

One of the biggest challenges of design exploration is that it can quickly generate dozens of experiments and the associated data. It would be a daunting task to sift through the large amount of data and filter out the important results. Thankfully, RTL Architect automatically examines the data and prepares two insightful summary reports.

The first report is an Explore Design summary of each experiment, as seen in figure 3, which provides all the key PPA metrics by category: timing, instance count, area, power, and congestion. In addition, a snapshot of each experiment's floor plan is provided. The detailed reports for each run can be examined by clicking on the run link for in-depth investigation.

Explore Design Summary																						
Run	Timing						Instance Count				Instance Area			Power						Congestion		
	WNS	TNS	NVE	Trans Cost	Trans Violation Num	Cap Violation Num	Standard	Register	Repeaters	ICG's	Standard	Register	Repeaters	Total	Leakage	Switching	LVTCell	LVTCell%	Image	#Cells in Congested Areas	MaxOverflow	GRC%
util60_aspect1-1	-0.32	-18.44	654	0.00	0	313	6855822	738938	886521	13550	1194623.06	3234442.04	83647.18	7.67e-02	3.46e-04	3.69e-02	0	0.00		9542	41	34164 (0.09%)
util70_aspect1-1	-0.22	-31.62	607	0.04	1	276	6848603	738938	891974	13550	1190212.07	3233983.11	83909.66	7.47e-02	3.47e-04	3.52e-02	0	0.00		10257	47	41294 (0.13%)
util60_aspect1-2	-0.87	-77.50	1259	1.05	4	289	6806446	738938	860368	13550	1182799.41	3233986.43	79484.60	7.51e-02	3.47e-04	3.53e-02	0	0.00		7838	64	30934 (0.08%)
util70_aspect1-2	-0.85	-96.11	1228	0.37	2	357	6823050	738938	886580	13550	1185512.68	3233997.93	84113.97	7.50e-02	3.48e-04	3.53e-02	0	0.00		18793	64	42073 (0.13%)

Figure 3: Explore Design Summary Report

The second report, as seen in figure 4, is a QoR comparison of each experiment which ranks the key PPA metrics. One run is designated as the reference and it appears in gold in the table. PPA metrics from the other runs are colored red if they are worse than the reference run, and the metrics are colored green if they are better than reference. This side-by-side comparison helps designers choose parameters, at-a-glance, that give them the desired power, performance, and area. In addition to the high-level QoR Summary comparison, QoR comparison reports can be viewed by category by selecting the metric of interest in the left-most column, such as, Power Summary.

QoRsum																						
QoR Summary	Timing						Instance Count				Instance Area			Power						Congestion		
	WNS	TNS	NVE	Trans Cost	Trans Violation Num	Cap Violation Num	Standard	Register	Repeaters	ICG's	Standard	Register	Repeaters	Total	Leakage	Switching	LVTCell	LVTCell%	Image	#Cells in Congested Areas	MaxOverflow	GRC%
util60_aspect1-2_layerM3	-0.40	-4.5	25	0.27	0.9	15	0	0	82.2	1638	6316	0	1520000	126000	2.5	1.00	18.10					
util60_aspect1-2_layerM2	-0.40	-4.5	25	0.27	0.9	15	0	0	82.2	1638	6316	0	1520000	126000	2.5	1.00	18.10					
util60_aspect1-2_layerM4	-0.40	-4.5	25	0.27	0.9	15	0	0	82.2	1638	6316	0	1520000	126000	2.5	1.00	18.10					
util60_aspect2-1_layerM4	-0.40	-4.5	25	0.27	0.9	15	0	0	82.2	1638	6316	0	1520000	126000	2.5	1.00	18.10					

Choose Baseline and Visible Runs

Run ID	Run Name	Baseline Run	Visible Runs (Summary)	Visible Runs (Detailed)
1	util60_aspect1-2_layerM3	☒	☒	☒
2	util60_aspect2-1_layerM3	☒	☒	☒
3	util60_aspect2-1_layerM4	☒	☒	☒
4	util60_aspect2-1_layerM5	☒	☒	☒

Check All Uncheck All

All checked runs will be shown in summary tables, but detailed tables show a maximum of six (the baseline and first five test runs).

Figure 4: QoR Comparison Report

Auto Floor Plan Advantage

The automatic floor plan creation capability is a key component of PRE. It enables the designer to do multiple what-if experiments in parallel without the burden of creating individual floor plans for each one. Even if the floor plan is not used in the final physical implementation, the RTL designer will obtain PPA results that are based on an actual physical implementation.

RTL Architect begins by creating a chip boundary using the default aspect ratio and cell utilization constraints, which is typically 1:1 and 60% respectively. The exact die size is determined from the standard cell and macro cell area. Then the IOs, standard cells and macros are placed in a timing and congestion driven manner. For hierarchical designs, it will place and shape hierarchical blocks before placing standard cells and macros. Blocks pins will be created and placed based on the global route. In addition, the RTL designer can supply default settings, pin constraints, macro groupings and even a floor plan Design Exchange Format (DEF) to drive the automatic floor planning flow.

Utilization Sweeps

A common concern for advanced node designs is wire congestion. This occurs when too many wires pass through an area of the chip. The Global Route Congestion (GRC) metric measures the congestion by calculating the ratio of the number of wires passing through an area of the chip, to the maximum capacity of that area. Ultimately, cell placement and floor plan configuration determine the amount of congestion. High fanout cells placed close together can cause congestion hotspots. Decreasing the cell utilization by growing the chip alleviates the congestion by allowing cells to spread to less congested areas. However, increasing it too much leads to unnecessary spreading and increases the overall chip area leading to less die per wafer. RTL Architect can perform a utilization sweep at a given aspect ratio, using the built-in options, to determine the largest chip size needed to alleviate the GRC. In fact, as mentioned earlier in the article, RTL Architect can perform a double sweep over a range of utilizations and aspect ratios. Changing the aspect to be taller or wider can increase horizontal or vertical routing resources, respectively. For each experiment, RTL Architect will analyze the design and indicate whether the congestion is caused by the logic or the layout. These analytics help the user to decide whether to restructure the RTL or modify the floor plan to improve the congestion. Figure 5 shows RTL Architect analyzing congestion and cross-probing to the RTL.

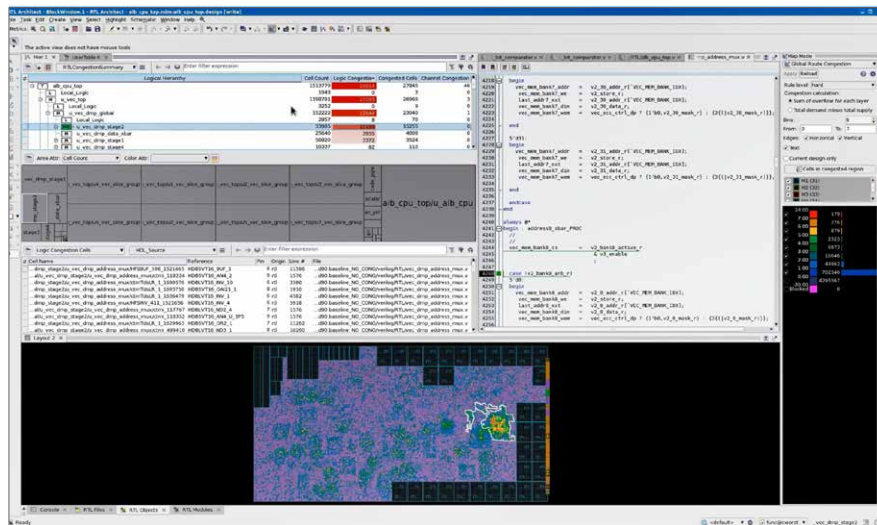


Figure 5: Congestion View—Cross-probing to RTL

Cell V_{th} Sweeps

Controlling power consumption is a primary concern for most chips, whether it is to extend battery life in mobile applications or reduce the cost of energy in a bitcoin mining operation. Choosing a standard cell library has a direct impact on the power. Chip manufacturers offer a variety of standard cell libraries for each technology node. The libraries are characterized by high, low, and nominal threshold voltages (V_{th}) required to “turn on/off” the transistor. Low V_{th} cells switch faster, so they have less cell delay which makes it easier for the path to meet setup timing requirements. The trade-off, however, is that the low V_{th} cells have higher leakage current which results in higher power dissipation. A V_{th} sweep allows the RTL designer to see the impact their code has on power and timing for a given mixture of library cells.

Figure 6 shows the experimental results of a four-run sweep. The first run allows only the Highest V_{th} cells to be used. Subsequent runs enable the next lowest V_{th} cells to be used, in succession. The Highest V_{th} run is used as the baseline, so power results are reported as a percent increase for easy comparison. Using only the Highest V_{th} cells results in almost 400ns of Total Negative Slack (TNS). TNS is the sum of all setup timing violations so it can indicate the effort needed to fix the violations in the place and route implementation stage. As lower V_{th} cells are enabled the TNS drops but the power increases. The sweep revealed that there was no benefit to TNS when the Lowest V_{th} cells are enabled, and the power is worse.

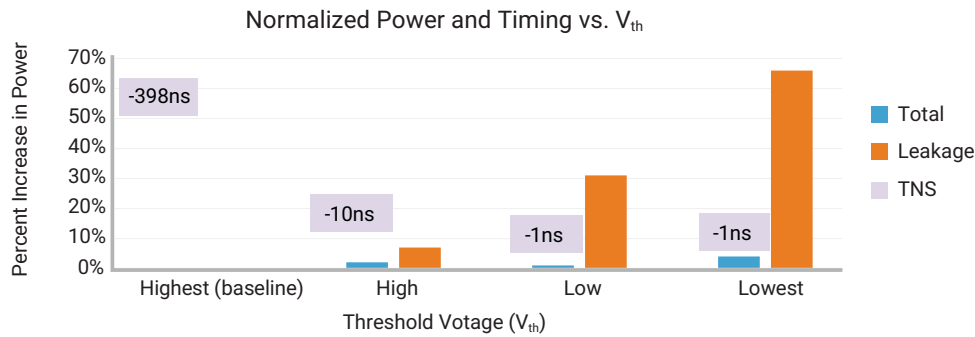


Figure 6: V_{th} Sweep for Power and Performance Tuning

Frequency and Voltage Sweeps

Fmax refers to the maximum clock frequency at which a design can run without setup timing path violations. The Worst Negative Slack (WNS) violation directly limits Fmax since the violation either must be fixed or the clock slowed down until the path meets the setup timing requirement. Importantly, Fmax also defines the maximum performance of the design which is the number of operations per second. High performance designs, like processors, want to increase Fmax to achieve the highest possible number of operations per second. Modern designs use dozens of clocks since portions of the chip operate at different frequencies, such as the interface logic. Certain clocks control the key engines that dictate the performance of the application specific chip.

It is useful for the RTL designer to sweep over different frequencies for these key clocks to see how fast they can go before causing setup timing violations. The size of the WNS indicates whether they are close to meeting timing or far off. If the WNS is large, the designer can then explore architecture changes that reduces the WNS which leads to faster clock speeds. Place and route implementation can fix smaller WNS violations. RTL Architect's design analytics helps the designer analyze WNS by module, and the trade-off in power since higher frequencies increases the dynamic power contribution to total power. Figure 7 shows the Module Timing Viewer in RTL Architect. It allows an RTL designer to see critical timing paths by module so they can identify which logic to change.

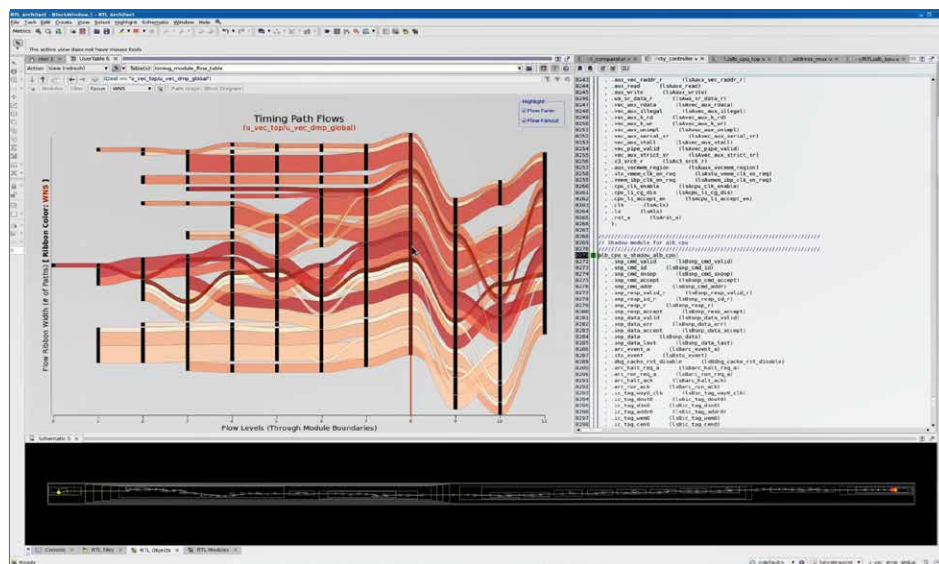


Figure 7: Module Timing View—Analyzing timing through an RTL module

After determining their V_{th} mix and tuning for Fmax, designers can perform library voltage scaling and frequency sweeps to generate V/F curves for Dynamic Voltage Frequency Scaling (DVFS). DVFS is a technique used to save power by dynamically reducing the voltage and clock frequency to portions of a chip based on task load. Voltage reduction has a bigger impact since the dynamic power consumption is proportional to CV^2f where C is capacitance, V is voltage, and f is frequency. RTL Architect can perform voltage and frequency sweeps with the user-defined-parameter feature. The user defines parameters for both the clock and voltage of interest in their run script. Then they pass a range of values to PRE which will launch multiple parallel runs. This is useful for predicting the V/F curve for a given V_{th} mix,

or analyzing the timing and power for pre-determined DVFS combinations. Figure 8 shows the results of a sweep of three frequencies at three different voltages. These nine experiments helped the designer predict performance at each DVFS points.

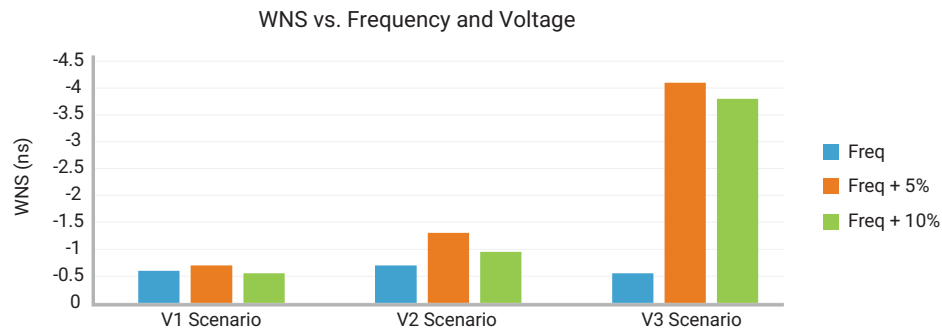


Figure 8: Timing at Various Voltage and Frequencies

RTL Sweeps

Finally, an RTL module owner may want to experiment with different versions of the RTL. For example, they may want to launch parallel runs with different configurations of a memory or restructure the RTL to combine modules to reduce cross-module communication congestion. The auto floor planner simplifies these experiments since it automatically creates a floor plan including the macro placement. Then they could use the QoR comparison utility to see the impact on timing, power, area, and floor plan congestion.

Conclusion

Parallel RTL Exploration offers RTL designers an automated, systematic approach to optimizing their designs for congestion, power, performance, and area. Furthermore, the programmatic interface, extensive data collection, and comparison utilities lays the foundation for AI-driven sweeps. The Synopsys Fusion Design Platform (Figure 9 Fusion Design Platform) continues to enable innovative integrated solutions developed to enable a new “shift-left” approach to solving PPA challenges at advanced process nodes. RTL Architect identifies and corrects physical implementation issues early in the design cycle to achieve aggressive PPA targets at advanced nodes through better RTL.

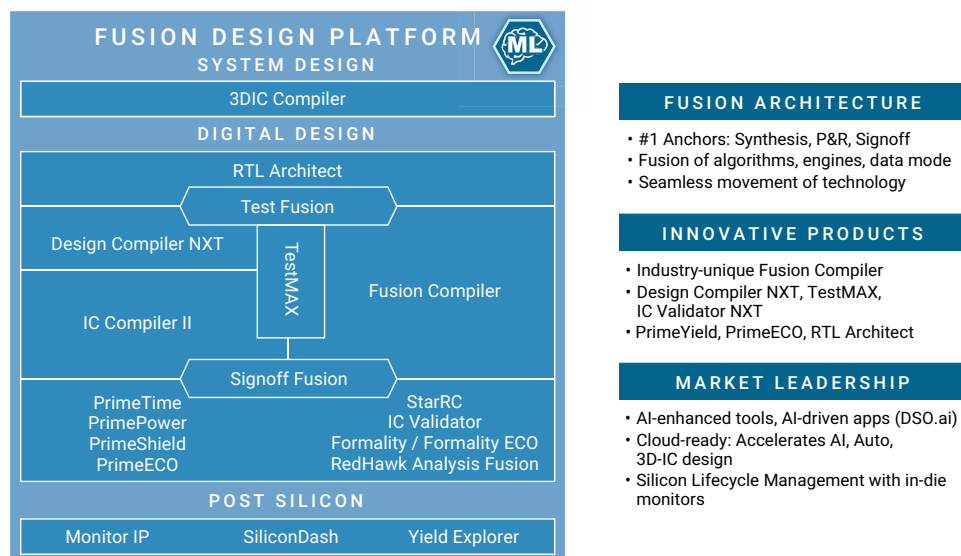


Figure 9: Fusion Design Platform

For more information, visit: <https://www.synopsys.com/fusion>