PRODUCT **BRIEF**

Genus Synthesis Solution

The future of digital is here

Synthesis Challenges with Complex, Advanced-Node SoCs

As system-on-chip (SoC) design size and complexity continue to rise and process nodes continue to shrink, the runtime and capacity of digital implementation tools have not been able to keep pace. At 16nm and below, there can be more than 500 physical blocks on an SoC, a more than 10X increase versus the 40nm node. The only way to manage this increase is to increase design team headcount, a practice that cannot continue indefinitely.

A key reason why implementation tools have not kept pace is that they currently cannot distribute synthesis, placement, and optimization effectively across multiple machines without degrading power, performance, and area (PPA). The highly iterative nature of digital design is another key factor. As process nodes shrink, increasing wire capacitance and resistance, more complex waveform propagation effects, and more complex physical design rules all compound to make it harder to correlate block-level synthesis to place and route (P&R) without dramatically increasing synthesis runtimes. These same challenges ripple back from block-level synthesis to "unit-level" synthesis environments run by individual logic designers, often resulting in almost no correlation between what an RTL designer sees at the unit level and what the physical design team sees in P&R.

As correlation from P&R to unit-level synthesis degrades, more and more iterations are required between unit-level synthesis, block-level synthesis, and P&R, adding yet more pressure to achieve a fast turnaround time of each of these tasks by keeping block sizes small.

A new generation of synthesis tool is needed to close this SoC "design productivity gap," one that is:

- Scalable and has the parallelism to handle a large volume of blocks
- Tightly correlated to P&R
- Capable of enabling unit-level physical synthesis with a full understanding of P&R at the block level

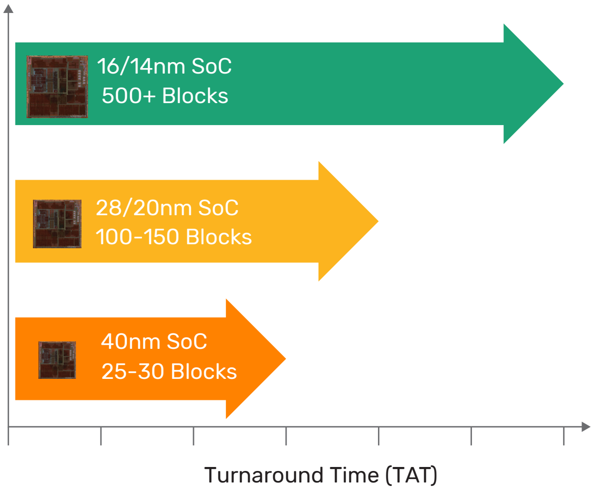


Figure 1: Increasing SoC complexity

Closing the Design Productivity Gap

Cadence has developed a next-generation logical and physical synthesis tool, the Genus™ Synthesis Solution, that is architected from the ground up to comprehensively address the design productivity gap. The solution is developed on a massively parallel architecture, delivers 3X-5X faster synthesis runtime, and is tightly correlated to the Cadence® Innovus™ Implementation System, which provides placement and routing. The solution also introduces a new global analytical algorithm for micro-architecture selection that can achieve dramatic reductions in datapath area without any impact on performance.

The architecture of the Genus Synthesis Solution allows it to scale well beyond 10 million instances. On designs up to one million instances, four to eight CPUs is typically sufficient. On designs up to 5 million instances, 8 to 32 CPUs work well. On designs of 10M+ instances, 64+ CPUs can be used. So long as each machine has four to eight CPUs available, the Genus Synthesis Solution is agnostic whether CPUs are on one machine or spread across multiple machines.

Massively Parallel Architecture

Synthesis tools today already leverage parallelism, but there are two key challenges which have limited the effectiveness of this parallelism to scale without degrading PPA:

The long pole effect. The number of gates distributed to each machine is not a good measure of optimization runtime. Different parts of a design may require different amounts of optimization effort to achieve a good result. If distribution is based only on gate count and not optimization effort, then some machine can become the “long pole” and hamper any runtime improvements from the distribution.

Design hierarchy. The best gate partitioning across machines to mitigate the long pole effect may not align with the user module hierarchy, so distribution needs to be able to “slice” through design hierarchy.

The Genus Synthesis Solution solves the second challenge with a low-level genetic change to its data model, which enables a design to be partitioned across module hierarchy. It solves the first challenge with a new algorithm that performs a proprietary timing-driven distribution of a design across multiple machines in such a way as to ensure that the optimization effort is equalized across each machine.

The distribution in the Genus Synthesis Solution is performed at two levels. First, there’s the coarse level on clusters of 100K or more instances. Second, there’s the fine level on clusters of 10K or more instances. A third level of algorithm-specific multi-threading is also performed on each machine within a shared memory space.

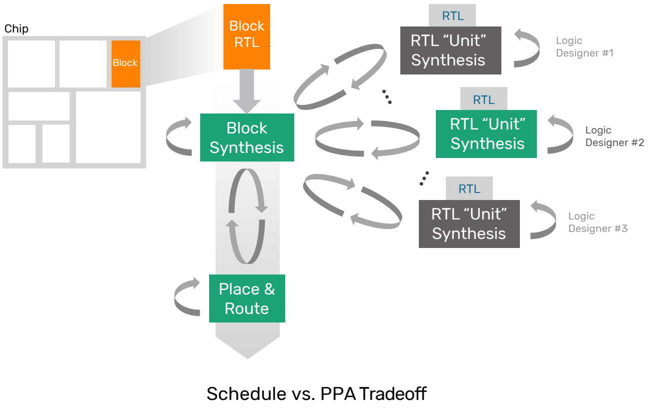


Figure 2: The iterative process of SoC design

With three levels of parallelism covering multiple machines and multiple CPUs per machine, it’s also vital to make sure that each level of parallelism is used in the right way at the right time to keep all CPUs on all machines utilized as much as possible. The Genus Synthesis Solution includes an intelligent scheduler for this purpose.

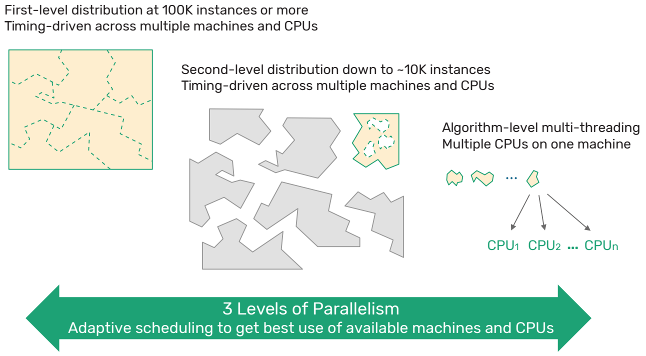

First-level distribution at 100K instances or more
Timing-driven across multiple machines and CPUs
Second-level distribution down to ~10K instances
Timing-driven across multiple machines and CPUs
Algorithm-level multi-threading
Multiple CPUs on one machine
CPU1 CPU2 ... CPUn
3 Levels of Parallelism
Adaptive scheduling to get best use of available machines and CPUs

Figure 3: Massively parallel architecture in Genus Synthesis Solution

Tight Correlation to P&R

Tight correlation between physical synthesis and P&R is easy if runtime is not a concern. The challenge is to achieve good correlation while maintaining the runtimes and capacity necessary to close the design productivity gap.

A key requirement for the Genus Synthesis Solution was to achieve tight correlation to Cadence’s new Innovus Implementation System without any increase in the total flow turnaround time for both solutions combined. The goal was for timing and wirelength at the end of Genus physical synthesis to correlate within 5% of the Innovus Implementation System and for Genus physical synthesis runtimes to be no more than 2X Genus logical synthesis runtimes.

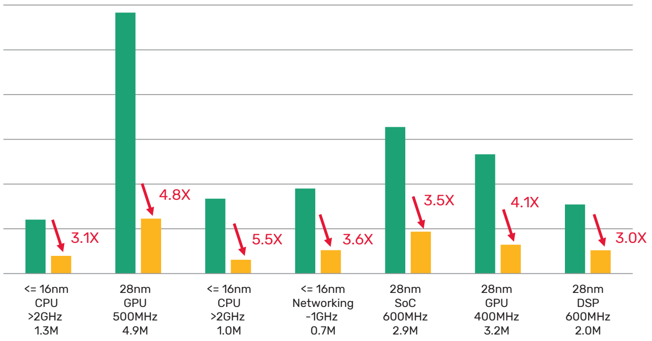


Figure 4: Genus runtime improvements

The only way to meet this requirement was through aggressive code and algorithm sharing throughout the Genus physical synthesis and Innovus placement optimization steps in the flow.

GigaPlace engine. The Genus Synthesis Solution leverages the same next-generation GigaPlace^{TM} technology used by the Innovus Implementation System. In a Genus physical synthesis flow, the initial placement is done in the Genus environment. Then, the Innovus system performs only an incremental physical optimization before clock tree synthesis.

Global routing. The Genus and Innovus solutions share a dramatically improved global router which is 4X faster but also delivers high-quality routes. It considers both wire capacitance and pin-to-pin wire resistance when constructing routes and has full industry-standard congestion control through net detouring and wire spreading.

Parasitic extraction and delay calculation. The Genus and Innovus solutions leverage unified parasitic extraction and delay calculation with full support for advanced-node waveform modeling.

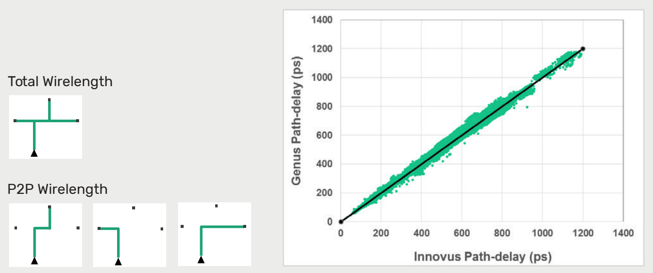
These unified engines also extend into the Cadence Tempus™ Timing Signoff Solution, enabling truly convergent front-to-back modeling through the full Cadence digital implementation flow.

Unified user interface. The Genus, Innovus, and Tempus solutions offer a fully unified Tcl scripting language and GUI environment. This unified user interface eliminates false perceptions of miscorrelation related to reporting format or data model query differences between tools. It also streamlines flow development and improves productivity of multi-tool users.

The Genus and Innovus solutions achieve total pin-to-pin wirelength correlation within just 1% and 3-sigma on total path delay correlation is within 5%. Frequency correlation is within 2%.

Timing-and Physically Aware Contexts

Tight correlation between P&R and synthesis reduces iterations between front-end and back-end design teams. A massively parallel distributed architecture enables block sizes to increase without increasing synthesis turnaround times. However, synthesis blocks are themselves typically formed by combining several RTL "units," each owned by a different logic designer.



	Encounter RTL Compiler Advanced Physical Option/ Encounter Digital Implementation System	Genus Innvovus Flow	Benefit
Frequency Delta	4%	2%	2X
Three-Sigma Path Delay Delta	8%	5%	1.5X
Total Wirelength Delta	2%	1%	2X
Total Pin-to-Pin Wirelength Delta	3%	1%	3X

Figure 5: Correlation between the Genus and Innovus environments

In addition to managing a centralized block-level synthesis environment that runs nightly or weekly, chip design teams typically generate "unit-level" synthesis environments that logic designers use during the day for debugging and tuning of their RTL. Traditionally, these unit-level environments are either purely logical without any modeling of physical effects, or they make very simplified assumptions about placement and floorplan that are independent of any physical synthesis at the block level.

To close the design productivity gap, it is important for logic designers to see a realistic view of PPA from their unit-level synthesis runs—one based on “clipping out” a true physical and timing context for their unit from the block-level physical synthesis environment.

Genus Synthesis Solution makes the creation of these unit-level “clips” very simple. With just one Tcl command at the end of a block-level synthesis script, the complete timing and physical context for any subset of a design can be extracted into a unit-level Genus database.

Genus clips include the full block-level floorplan, as well as placement and timing for all interface logic connecting to and from the unit. Complex overlapping clocks and all interface timing exceptions, such as false or multi-cycle paths, are modeled perfectly.

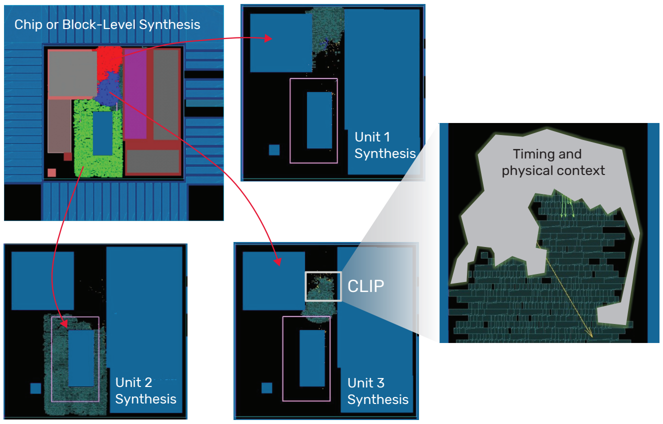


Figure 6: Timing- and physically aware contexts

Logic design teams can use Genus clips to iterate rapidly and accurately on their unit-level RTL, testing each change against the full-chip or block-level physical context. Anytime the system implementation engineer rebuilds physical blocks, the unit-level clips can be automatically refreshed. The result is a tight and accurate design loop, ensuring that day-to-day RTL coding decisions are made with full knowledge of the physical impact at the chip or block level.

Early customer feedback using Genus clips indicates an improvement in productivity of more than 2X through reduced iterations between unit- and block-level synthesis.

Global Analytical Architecture Optimization

For a given RTL description, there can often be many different possible micro-architectures, each with very different PPA tradeoffs. Since different architectures for key datapath regions in a design can have such a dramatic impact on PPA, picking the right datapath micro-architecture is one of the most important decisions in the overall digital implementation flow.

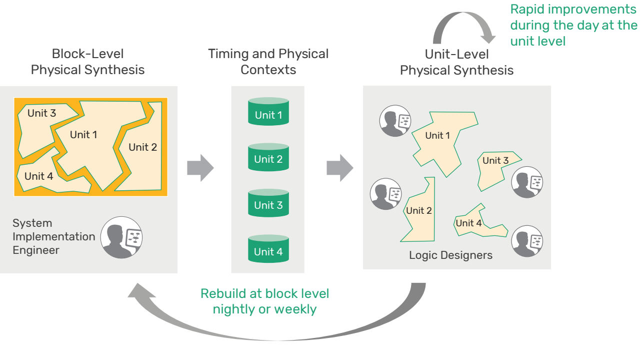

Block-Level Physical Synthesis
Timing and Physical Contexts
Unit-Level Physical Synthesis
Rapid improvements during the day at the unit level
Unit 3
Unit 1
Unit 2
Unit 4
System Implementation Engineer
Unit 1
Unit 2
Unit 3
Unit 4
Unit 1
Unit 3
Unit 2
Unit 4
Logic Designers
Rebuild at block level nightly or weekly

Figure 7: Contexts improve RTL design productivity

Traditional approaches to datapath architecture selection in synthesis typically begin with an area focus and then incrementally refine these architectures where timing is critical to meet frequency goals. This approach is runtime inefficient and can converge on a local optimum, which leaves significant PPA on the table.

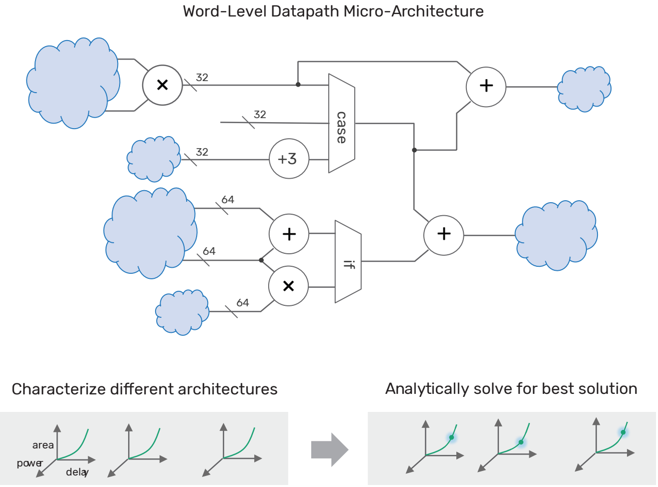

Word-Level Datapath Micro-Architecture
×
32
32
32
+3
case
+
64
64
64
+
×
if
+
Characterize different architectures
Analytically solve for best solution
area
power
delay

Figure 8: Global analytical architecture optimization

To achieve high-quality PPA with fast runtimes and high capacity, the Genus Synthesis Solution adopts a "right first time" approach to its micro-architecture selection. This approach begins by building a graph of all critical datapath regions in a design regardless of their physical or logical module hierarchy.

For each of these regions, the Genus Synthesis Solution considers a range of possible mirco-architectures with various PPA tradeoffs. It builds an analytical model for these different architectures, and solves the model globally across all datapath regions in the design for the overall best design-level PPA.

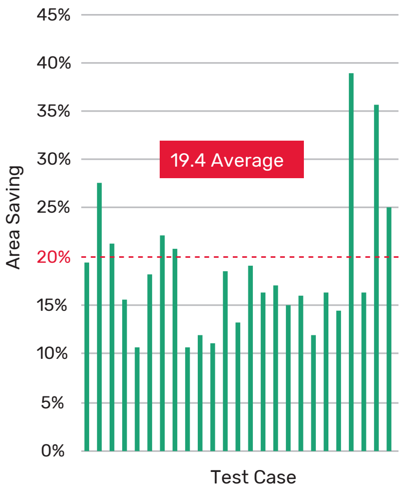


Figure 9: Area savings from global analytical datapath optimization

A global analytical approach to architecture optimization can achieve dramatic improvements in PPA that are not possible from a non-analytical, progressive refinement scheme. For example, on a testsuite of datapath-intensive design modules across a range of different SoC applications, an average area reduction of more than 19% is achieved without any impact on performance.

Summary

As SoC design size and complexity have continued to increase, the runtime and capacity of digital implementation tools have not kept pace. Shrinking process nodes have also eroded correlation between P&R, block-level synthesis, and unit-level synthesis, increasing iterations required to meet PPA goals and creating pressure to keep block sizes small.

Cadence's Genus Synthesis Solution was architected to support block sizes of 10 million instances or more and achieve excellent correlation between P&R, block- level synthesis, and unit-level synthesis. Its multi-level massively parallel architecture reliably achieves a 3-5X improvement in runtime. Extensive engine unification with the Innovus Implementation System enables Genus physical synthesis to achieve timing and wirelength correlation within 5% of the Innovus solution. Native support for full timing and physical context extraction reduces iterations between block- and unit-level synthesis by 2X or more, and a new global analytical algorithm for micro-architecture selection achieves up to a 20% reduction in datapath area without any impact in performance.