

Cracking Intractable Chip Design Challenges with the Help of Machine Learning

By Anirudh Devgan, Cadence 11.22.2020

Chip design is where the immutable meets the intractable. From some viewpoints, every element in a chip design is open to modification, but in practice there are consequences to every action. The need to track those effects is why electronic design automation (EDA) was invented, and once you start to analyze a chip design at that level you soon appreciate the complexity involved. With design sizes increasing, concurrently managing those parameters is more challenging. Interdependency on that scale means that a change to any parameter could have a negative impact on the power, performance and area (PPA) goals we set.



Anirudh Devgan, President of Cadence
(Image: Cadence)

At every step along the way, we use tools that address a specific part of the design process. Long ago, when things were comparatively much simpler, each stage would hand off to the next in a linear fashion, similar to the way runners complete a relay race. With advanced process nodes, there are many more parameters to consider and all decisions are co-dependent. This can be likened to running a race while discovering the course amidst obstacles, weather, traffic and other factors, after the starter pistol is fired.

Moving from a linear to a concurrent chip design flow hasn't happened randomly; it is born from necessity. That necessity comes from the complexity, not just the market pressures that demand new products at an ever-decreasing interval. The complexity comes, in part, from the continued pursuit of decreased feature sizes and increased levels of functionality, leading to higher overall integration. Increasingly, we are seeing greater configurability and flexibility at the chip, module and PCB level. Each component, whatever form it may take, comes with its own demands. We can think of it as object-oriented hardware design, as every component has parameters, every parameter is variable to some extent, and every variation has multiple connotations for the wider design.

It would be impossible, literally, to compute the “what if” scenarios for each of the possible variations in every one of those parameters for each and every component in a system, but that is effectively what we are now asking of the EDA tools that facilitate design. This is where forward visibility, enabled by what we call the digital full flow, fits. This is a concept that goes beyond how to deliver better tools for synthesis, or place and route; it is about the higher level drivers that are enabled by the digital era and computational software.

Something that is also becoming more apparent as the semiconductor industry continues to evolve is that what may have seemed small and insignificant in the past is now becoming comparatively large and imposing. There is no “ideal world”, so IC designers have always had to keep one eye on system parameters, such as thermal management requirements and signal integrity issues. These system parameters can easily become problematic if left unchecked. Today, those parameters need even closer consideration and often become design limitations in their own right. Modern chip design teams need to monitor more of the system, more of the time. We feel this can only be addressed using an approach to design founded on a distributed compute environment, with tools that have been redesigned to operate as part of a massively parallel architecture. The introduction of machine learning (ML) into EDA tools also provides a relief valve for designers. Now, ML can take on the task of checking and even actually designing large portions of the system. By assuming more of the design effort, ML alerts the designers when it recognizes that something has the potential to become a problem.

In this way, ML is used to deliver the forward visibility mentioned earlier, by inferring PPA results earlier in the design flow using the knowledge enabled by its ability to learn. This level of change happens at the very lowest level, demanding entirely new algorithms in the tools that support not only ML but a much more concurrent workflow.

This seismic shift is necessary to deliver a deployable and scalable tool flow that can be applied to larger designs at nodes of 16nm and below. If there is one word that could begin to encapsulate this, it might be “correlation”, as at every stage there is correlation between design data. This isn’t simply about functional implementation; it extends to every aspect of system performance and every tool used to manage those aspects. These all need to work together, in the digital full flow, if they are to work at all. In practice, this is enabled by a reimagining of how things happen under the hood. More of the tools in the flow now use ML to handle the vast amount of low-level data that the tools generate. The wider use of common and shared engines distributed across more processing resources means that there is greater correlation of the higher level design data throughout the entire flow.

The EDA landscape shifts constantly; it must in order to support the movement of the semiconductor industry. There has always been an advantage gained from close correlation between the performance of the tools and the needs of the end markets. Now, the advantage comes from having ML in a digital full flow that enables the correlation of all design data, as it moves without restriction around the EDA domain.