

# Faster and Better Floorplanning with ML-Based Macro Placement

## Author

**Jim Schultz**

Senior Product Manager,  
Synopsys

## Introduction

The chips contained in today's consumer and commercial electronic products are staggering in size and complexity. The largest devices include central processing units (CPUs), graphics processing units (GPUs), and system-on-chip (SoC) devices that integrate many functions on a single die. Additionally, chips are expanding beyond their traditional borders with multi-die approaches such as 2.5DIC and 3DIC as a way to improve data transfer and mix technology nodes, which will propel applications like autonomous vehicles to improved efficiency and cost.

One thing that all these leading-edge technologies have in common is a huge number of design elements. Each die may contain billions of transistors and millions of instances, split between memory arrays and logic functions. This complexity is required to satisfy the demands of emerging markets such as artificial intelligence (AI), high-performance computing (HPC), and hyperscale data centers. Meeting the power, performance, area, and congestion (PPAC) targets for these designs is increasingly challenging, consuming a great deal of project resources and lengthening time to market (TTM).

This white paper focuses chip floorplanning and the key step of macro placement, which is crucial for satisfying PPAC requirements: Similar to building a house or a skyscraper, a floorplan is a blueprint to build a chip that delivers the desired PPAC results. As in so many areas of technology, AI-based techniques such as machine learning (ML) can play a big role in taking floorplanning to the next level.

## Floorplanning Boot Camp

Theoretically, no matter how many placeable instances a design may have, the physical design team can submit the entire netlist into the place-and-route (P&R) process and get clean results. For small designs, it is possible for a layout tool to place all the memory macros and standard cells without a blueprint, then route all the signal nets that interconnect them. In practice, however, the runtime and compute requirements to implement large chips become impractical without a floorplan.

The problem is that the resulting layout may fail to meet PPAC requirements. Post-route static timing analysis (STA) often reveals paths that exceed the designed cycle time, compromising performance. The die area may be too large to meet product cost targets, or the power consumption may make the chip unsuitable for its intended end use. Congestion caused by densely packed cells could make some nets un-routable. Tweaking the constraints and rerunning the whole P&R process for large designs is tedious and time-consuming. Many iterations may be required, each possibly taking weeks to complete and terabytes of disk space. Even then, the design may not ultimately meet its PPAC goals.

Floorplanning emerged as a way to help. Floorplanning is the act of physically constraining the components of a chip (standard cells, memory macros, and wires). It typically involves grouping related logic and constraining its placement to specific areas of the chip. Grouping related logic typically reduces the connectivity wire length, which results in improved performance. The assumption is that the chip designers know a lot about how the different parts of the chip interact, so they can guide a layout tool's place and route engines. A high-quality floorplan helps P&R converge faster to its PPAC targets.

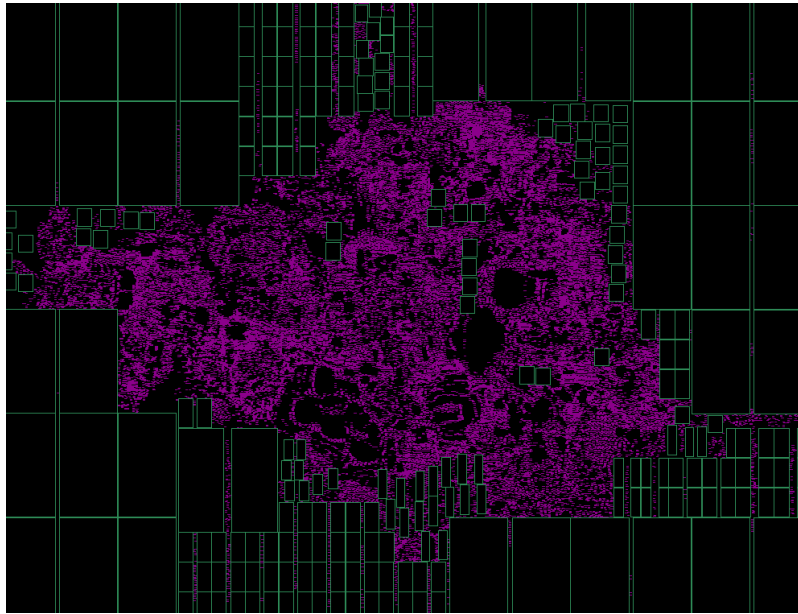


Figure 1: Example of a complex chip floorplan

## Limitations of Traditional Methods

Floorplanning is an important step in the physical design flow and it works very well for small to medium-sized designs. However, in modern large chips it too has become tedious, time-consuming, and iterative in nature. The challenges start with the partitioning process. Typically, when designs have more than 10 million instances, they are broken up into separate blocks in a process called hierarchical partitioning. This enables blocks to be independently and concurrently placed and routed, reducing the overall runtime. The completed blocks are then assembled to form a complete chip. This hierarchical implementation flow requires careful floorplanning so that the assembled blocks meet all the PPAC targets and pass all the design rule checks.

Traditional floorplanning often entails manual trial and error to achieve a good data flow for the chip. Placement of macros is a critical step in floorplanning because they are much larger than standard cells and their wide bus interfaces with many connections heavily influence routing congestion. Once the macros have been optimally placed to reduce wirelength and congestion, the remaining space is devoted to the standard cells. The goal is to place the macros and cells in each block in a manner that makes it most likely to meet the PPAC targets for the project. On familiar designs, floorplan creators might be able to rely on their experience and some institutional knowledge to speed up the process. For example, an expert on SoCs for HPC designs might have a sense for how to place and route the chip to extract the performance needed while minimizing power consumption and avoiding route congestion.

With growing chip size and complexity, the number of memories and other hard macros in a design is also rapidly increasing. When the number of macros grows into the thousands, limits how quickly manual efforts can proceed. As a result, floorplan design is dominating project schedules, and designers are looking for ways to meet their aggressive quality-of-results (QoR) goals with a reduced number of floorplan iterations.

There are many challenges to effective floorplanning of modern chips. Advanced rules for finer geometries (boundary cell, end of line, layer coloring, via enclosure, etc.), add to the complexity of placement and routing. As designers race to add new functionality, they often fail to plan for the increased power and area requirements, which can lead to costly late-stage re-floorplanning. Adding structures for test, safety, and security also puts stress on area and power, making floorplanning both more critical and harder.

Chip limitations that affect floorplanning include architectural requirements, gate counts, power modes, and power domains. Most chips use multiple power supply voltages that depend on the logic requirements. For example, a CPU voltage might be higher than USB or PCIe controller. This requires logic cells to be grouped and placed by power domain so they can receive the correct power supply routing. Additionally, power domains are turned on and off to conserve power, which leads to additional floorplan requirements to account for the insertion of power switches and level shifters. The bottom line is that manual floorplan design iterations can span from days to weeks depending on the size and complexity of the chip.

## Better Floorplanning with Machine Learning

Increasing design complexity requires elevating the crucial floorplanning step with automation. Automation can reduce iterations, shrink floorplan design time, and accelerate tapeout schedules. Machine learning automation provides the power and intelligence needed to meet the demanding PPAC targets for today's chips. As shown in Figure 2, ML-based floorplanning performs on-the-fly placement explorations much faster than any manual process could ever achieve. These "what-if" experiments can iterate rapidly through many possible floorplanning approaches.

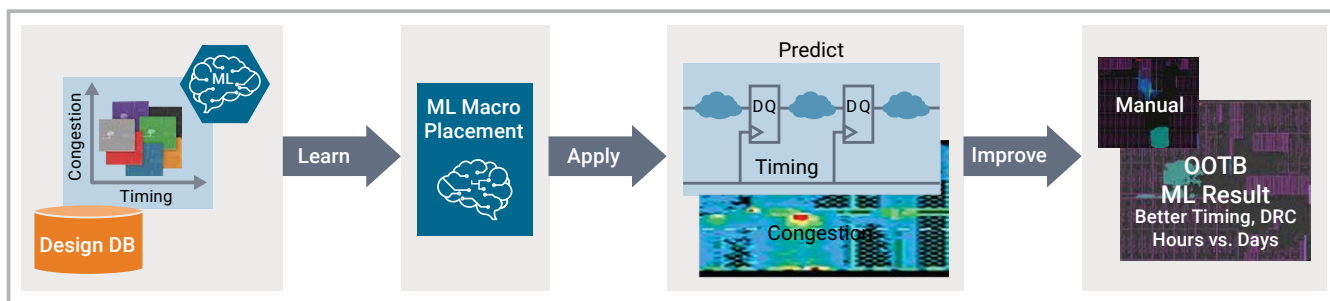


Figure 2: Applying ML to improve floorplanning.

ML doesn't just perform these trial placements; it learns from them as well. Layouts that produce inferior results are quickly discarded and the algorithms converge on those with the most promise. By automatically exploring hundreds of floorplans on-the-fly, the technology can generate the top-performing floorplan output. ML models are trained along the way, and the more data available for the training, the smarter the technology becomes over time. Given the vast exploration space in a large chip, particularly AI architectures that commonly include thousands of macros, ML techniques are well suited to address the challenges of floorplan design. Factory ML data in the library and the accumulated ML data from use on the project are saved for reuse by other projects, especially similar or derivative designs.

ML technology predicts congestion, wirelength, power, and total negative slack (TNS), producing a floorplan superior to manual methods. After place and route, the resulting layout is much better optimized for PPAC goals than layouts based on manual floorplans. This is not a one-time savings; as the design evolves over the course of the project the floorplan also evolves, and the place-and-route step is rerun many times. A great deal of designer work is eliminated on each iteration and the manual tuning effort is greatly reduced.

## The Synopsys Solution

Synopsys has made significant investments in AI and ML algorithms. In particular, the Machine Learning Macro Placement (MLMP) technology address a key challenge of traditional manual floorplanning. The MLMP solution automates macro placement iterations while reducing experiments via machine learning. It quickly searches a large solution space to find the best layout that gives the most optimized PPAC.

The macro placement engine supports multiple styles of placement: on-edge, free-form, and hybrid. The on-edge style, stacks macros around the edges of the chip to leave a large empty area in the middle for standard cell placement to reduce congestion. The free-form style allows macros to be placed in the middle near related logic to reduce wirelength to improve timing and power. The hybrid style enables the tool to intelligently choose between on-edge and free-form styles.

The ML technology generates a very large number of floorplan experiments with different macro placement solutions. A small subset of these experiments is taken through the full P&R flow to generate QoR data and to further train the Factory ML model. The trained model is then used to predict the QoR of the large number of experiments and identify the floorplan that produces the best PPAC creating the most optimal out-of-the-box (OOTB) macro placement for congestion and timing. This eliminates a great deal of manual time and effort at multiple points in the chip project. In addition, the high degree of automation means that designers can use the solution effectively with minimal training.

The results, when compared to traditional methods, are compelling. Figure 3 summarizes just a few of the measurements made by end users on real-world chip projects. These span a wide range of advanced applications, including AI, 5G, and the Arm DynamIQ Shared Unit (DSU) for interconnecting multiple CPUs. Results were improved for many design metrics, including TNS, worst negative slack (WNS), leakage power, engineering change order (ECO) loops, maximum operating frequency (Fmax), and schedule time.

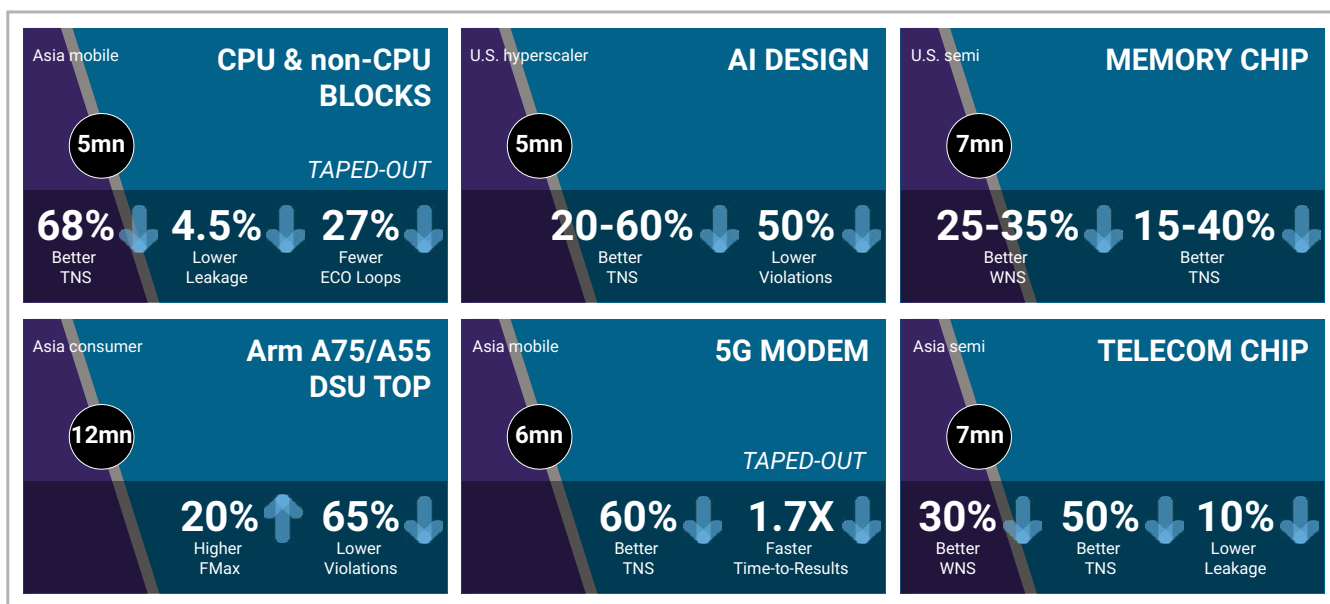


Figure 3: Real-world results of MLMP versus manual floorplanning

In addition, Synopsys DSO.ai complements MLMP by considering many more design options than just macro placement style. The additional design options are referred to as permutons and they can further improve the QoR by exploring design variations beyond macro placement, such as voltage scaling and library optimization. DSO.ai gets smarter with each iteration which enables it to converge faster on PPAC targets. These warm starts save time and resources.

The Machine Learning Macro Placement technology is available today in the Synopsys IC Compiler™ II and Synopsys Fusion Compiler™ P&R solutions, which bring automation and intelligence to the layout process. Synopsys IC Compiler II, Synopsys Fusion Compiler, and Synopsys DSO.ai are part of the Synopsys Digital Design Family, the industry's first AI-enhanced, cloud-ready design solution set that redefines conventional electronic design automation (EDA) tool boundaries across synthesis, P&R, and signoff. This comprehensive platform is geared toward delivering optimal PPAC and time-to-results.

## Summary

Layout and timing results for complex chip and multi-die designs require a floorplan to map out where essential components should be placed. The goal of a floorplan is to place macros and standard cells in a way that supports good data flow for the chip, to generate the best PPAC for the target application. Traditional manual floorplanning is an iterative, time-consuming, and resource-intensive process. Synopsys provides a new, automated, ML-driven technology along with DSO.ai to streamline the floorplan design process for improved productivity and produce results up to 70% better as measured on industry designs. Using the latest floorplanning technology designed with automation and intelligence can generate both superior QoR and optimal PPAC while meeting demanding TTM schedules.