

$$\nabla \times \vec{H} = j\omega\epsilon\vec{E} + \vec{J}$$

$$\dot{X}(t) = f(X(t), u(t))$$

$$\nabla \cdot (\epsilon\vec{E}) = \rho$$

$$\nabla \times \frac{1}{\mu} \nabla \times \vec{E} + j\omega\sigma\vec{E} - \omega^2\epsilon\vec{E} = -j\omega\vec{J}$$

$$\Delta t_k = \frac{-\dot{X}_k}{\ddot{X}_k}$$

$$\dot{X}(t) = \frac{\partial f}{\partial X} X + \frac{\partial f}{\partial u} u$$

Computational Software: Foundation for an Optimal Digital Design and Signoff Full Flow

By Chin-Chi Teng, Senior Vice President and General Manager at Cadence

Cadence has started to use the term, “computational software”, to collectively refer to a superset of algorithms and technologies of those powering electronic design automation (EDA) tools. Providing the most powerful computational software for EDA is probably most applicable in the business group I’m responsible for: Digital Design and Signoff. We have to enable our customers to handle the biggest designs with the most objects whether we are measuring in transistors, gates or placeable objects, often navigating through the longest runtimes. If you work on these leading-edge designs, you already know that when we say that “physical design might take a week”, this is not a figure of speech. Not only that, but it may also require use of the most powerful servers in existence, too.

In the last couple of years as Cadence has gotten more heavily involved with the commercial cloud companies, I’ve discovered that they are surprised when they discover that we mean it when we say, “run for a week”. Nothing other than EDA in general, and especially digital physical design and signoff, comes close. For example, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) contains 1,281,167 images for training, 50,000 for validation, and 100,000 for testing. By EDA standards, these are small numbers.

There are multiple computational software engines involved in the digital flow—engines for synthesis, placement, clock-tree, routing, timing, extraction, and power. These all need to be best in class to get a best-in-class result. In the dim and distant past, these engines would run one after the other (first synthesis, then placement, then clock tree...and so on). That is not accurate enough for all the physical details that need to be correct in a modern leading-edge process like 16nm and below, especially at the current cutting edge of 5nm and below. There also needs to be world-class integration so that, for example, placement can be taken into account and partially completed during synthesis, or so synthesis can be used during physical design to restructure logic.

Integration

Integration between synthesis and placement to improve design optimization is so important that we gave this technology a name, which we call iSpatial. Interconnect layers in a modern process all have very different capacitance and resistance profiles, so deciding how to use them must be done early and involves techniques such as layer assignment, useful clock skew and via pillars. The iSpatial technology allows a seamless transition from the Cadence® Genus Synthesis Solution to the Cadence Innovus Implementation System using a common integrated physical optimization, common user interfaces (UIs) and a common database exchange.

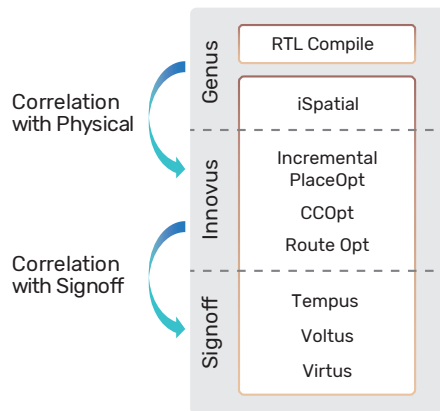


Figure 1: Digital implementation integrated flow

When a design requires a week of runtime, it is very important. Saving 10% saves about a day. A reduction like this can result from all sorts of reasons: Better scalability in the cloud (being able to use 48 machines instead of 32), better algorithms (a faster placer) or better integration (for example, between the placement engine and the timing engine).

Over the last several years, Cadence has created a digital full flow with common integrated engines, driving technology leadership. Of course, development is never over, but this gives us the platform to move into the future. Advanced nodes—5nm, 3nm, and beyond—are still very important to our most advanced customers.

More than Moore

One electronics industry trend we’re seeing relates to advanced packaging, often named with the catchy term, “More than Moore”. Integrating everything on the most advanced node is often not the most cost-effective approach, and it makes a lot of sense to use the most advanced process node only for the heart-of-the-system-on-chip (SoC) that requires it while using a less aggressive node to interface to the outside world. Not only does this approach yield better, but it can also result in a faster time to market since the current chip can use the input/output (I/O) designs from the previous process generation, and the current-generation I/Os can be developed with their test chips off the critical path. They will then be ready for the next-generation chip, perhaps a year later.

At one level, our customers are rushing to the most advanced nodes as fast as the foundries can deliver them. But they are being more judicious than in the past to ensure that they use these most advanced nodes where it gives them true differentiation. More than Moore has changed the tradeoffs in many ways. Only a few years ago, if you went to the next node, you moved everything to the next node and

integrated it all on a big SoC, with the possible exception of high-bandwidth memory (HBM) or package-on-package (PoP) dynamic random access memory (DRAM).

More than Moore is driving an even larger computational software challenge that requires simultaneous multi-chip digital design and signoff. Furthermore, grouping multiple advanced-node devices into a single package requires complex power and thermal analysis of the system to ensure it will function across all operating conditions. This increases the complexity of the chip designers’ task, which also presents a clear case for a robust digital design and signoff platform.

Machine Learning

Another computational software trend is to use machine learning (ML) to steer the underlying algorithms and thus improve results. One of the limiting factors in any advanced SoC design is having enough designers available. In some ways, running EDA tools has something in common with running a nuclear power plant: A lot of it is very repetitive, but you absolutely need the best engineers. ML allows computation, especially in the cloud, to substitute for routine human interaction, and thus, it can drive a major increase in productivity. In much the same way as a goal for cars is autonomous driving, but we have to get there through incremental steps. The goal for the digital full flow is eventual front-to-back automation, sometimes called, “no human in the loop”. But, as with cars, we have to approach through incremental steps—perhaps we can call that, “fewer humans in the loop”. When an engineer is running a tool, looking at the result, and then tweaking some parameters before running the flow again, there are a lot of opportunities to tweak the parameters automatically, at least some of the time. We can also capture the decisions of the best designers, and thus deliver some of their expertise to less experienced teams. It enables an engineering team to get more done.

Predictability

There is also plenty of scope for improving predictability. At some level, each tool in the flow has a standard of “goodness” that is tied up in how well it integrates with the next stage. A good placement of the design is one that the global router handles well. A good global route of the design is one that the detail router can handle well, and so on. Almost all the algorithms in EDA are computationally intractable in the sense that getting the exact optimal solution is not possible. Instead, heuristics are used. But this is another area where ML can be used, with heuristics at one level using ML to better predict how things will be downstream.

Better prediction leads to both a better result, and, at least potentially, a faster conclusion due to the reduction in the

amount of iteration required. Diagrams of a flow always look much more linear than they really are, since there is so much iteration going on both under the hood of most tools and in the flow with which the tool is repeatedly invoked.

Digital Full Flow

All of this reminds me of a remark by a comedian who said, "it takes a lot of practice to make it look unrehearsed". In the digital full flow, it takes a lot of computation and iteration to make it look smooth and linear.

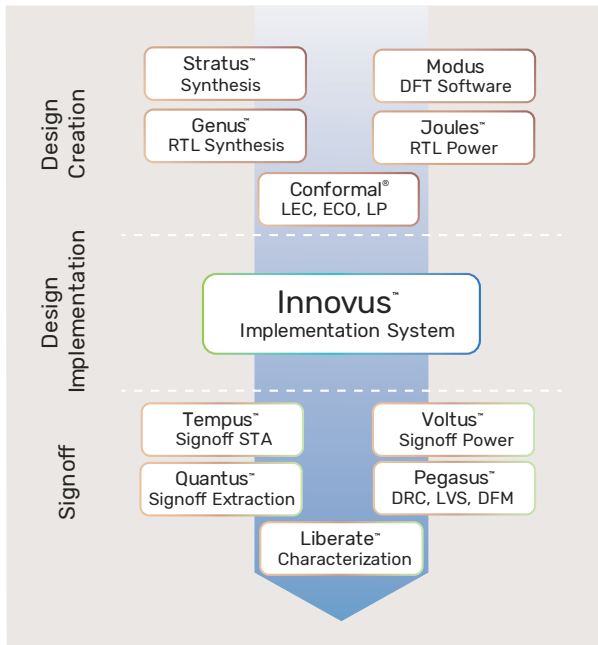


Figure 2: Cadence digital full-flow

The Cadence digital full-flow contains best-in-class engines, together with best-in-class integration to make everything work cleanly together. Adding ML improves productivity even more. You need best-in-class digital computational software to create and sign off best-in-class results.

Let me give you an example of best-in-class results from a GM at one of the top 10 semiconductor companies, who is in charge of CPU and AI design, that highlights his experience using our latest ML capabilities to train a model of their CPU core. The project resulted in both an improved maximum clock frequency and stunning 80% reduction in total negative slack. That's a bit of a geeky measure that might not mean much. Let me say that it's a big number. But it's not just a number—it resulted in a 2X shorter turnaround time for final signoff design closure.

To wrap up, the scale of computational software that is used in the digital and signoff flow is orders of magnitude larger than algorithms in other industries. The fact that people can design systems with billions of transistors is the most obvious evidence of that. The largest planes in the world contain about four million parts. Compared to digital implementation, that is a small number.

Visit www.cadence.com/en_US/home/company/computational-software.html to learn more.

