



BY PALLAB CHATTERJEE, CONTRIBUTING TECHNICAL EDITOR

Pcell and IP realities for fabless design

The current methodology for SOC (system-on-chip) design employs a hierarchical approach that maximizes the use of IP (intellectual-property) blocks. This method replaces the previous one, which used flat files full of gate-level structures. Third-party IP providers and in-house groups supply these IP blocks. Most of the in-house-created IP blocks are hierarchical assemblages of primitive devices that automatic cell generators, such as Pcells, create.

For large IDMs (integrated-device manufacturers) and large companies, this model works well. But to use this methodology, a company needs a full-time engineering and CAD (computer-aided-design) staff keeping the Pcell generators synchronized. The company also requires in-house macro design and characterization, as well as partnership arrangements with IP providers so that the IP remains current with the process rules and characterization. These staffing resources are in addition to the CAD support staff's installing and maintaining the EDA tools in the design flow and providing all the stitch-and-launch code interfacing between tools.

Smaller companies generally do not have these luxuries. The limited manpower resources at most small companies and the need to time-share that manpower over multiple tasks in the course of a design project make these luxuries infeasible. These limitations drive the smaller fabless companies to rely on PDK (process-design-kit) packages from the foundries as the basis for all information. For most sub-wavelength processes of less than 180

nm, these PDKs have design rules and model changes on a quarterly, monthly, and sometimes even weekly basis.

As a result of the constant changes in PDKs, most companies have time only for validating and correcting their custom IP blocks. For this task, downloadable technology files and verification files provide the most definitive information on the yield and reliability parameters of the process. Most print documentation is not as current as electronic files. A statistical-design approach suits systematic-design validation, allowing changes to individual parameters without context.

But the lack of resources to keep everything current creates other serious problems for smaller design teams. From a physical-verification perspective, the major challenge occurs when the design team begins point-rule optimization for yield enhancement. The quantity of interrelated design rules complicates this task, resulting in a set of rules that does not track.

The use of I/O cells from third parties and foundries creates another challenge. These I/O cells tend to be more specialized than the core-library device

es in the foundry's eyes, and the IP provider develops these cells early in the process life cycle. As a result, these devices do not incorporate ongoing process enhancements, as the IP supplier updates them only every few years.

The last revision issue is with the coding for the Pcells or auto-device generators. IP groups create Pcell-generator code for new processes mostly by modifying previous-generation legacy Pcell code. The method of modifying legacy code brings with it assumptions, work-arounds, patches, and parameters that were originally created to solve anomalous problems in previous process nodes, and these anomalies may not exist or be relevant for the current node. This method also limits the Pcell environment to using the process parameters and relationships that were available at the time of code migration and not incorporating the latest manufacturing enhancements. A large number of the Pcells, despite having geometrically accurate coding, create non-manufacturable structures, such as donuts with no construction line, off-grid vertices, and self-intersecting structures. As a result, correctly using these Pcells requires converting most of them to hard cells and then modifying them to meet current PDK requirements.

Using the structured-hierarchy methodology for SOC construction is a good and valid approach. However, most companies must allocate significant resources to PDK support, Pcell and custom-design support, in-house macro-revision control, third-party IP-revision control and waiver tracking, and I/O-cell-revision control and waiver tracking. Subwavelength-design flows have a significantly higher cost—both in manufacturing dollars and resources—than large-geometry processes. Most companies see only the manufacturing costs, but the engineering-resource allocation is what keeps most projects from completing on schedule. **EDN**

Contact me at pallabc@siliconmap.net.