# Datamodels for Physical Synthesis:
# Some Practical Considerations for Library Support

Dwight Hill and Shir-Shen Chang

Abstract for Electronic Design Processes Symposium, 2001

Physical Synthesis is the EDA area that merges the traditional functions of Physical Design, such as floorplanning and placement, with those of logic synthesis. Each of these areas has many stringent and complex data model dependencies. Combining them together poses problems for a data model that are not well addressed by tools that address either side in isolation.

One aspect of this that is generally not well covered in the literature is support for libraries, and the way that library data is linked with design specific data. Specifically, a Physical Synthesis data model must include support for libraries from both the logical and physical side. Logical library models must include functional models (for synthesis, IPO, etc), timing models, and testing. Physical models must include the cell geometries, and physical technology data, such as wiring design rules, via types, and extraction parameters. Complexity arises from several directions

- The logical, timing, physical and process libraries tend to be expressed in disjoint formats, and come from several very distinct sources. Even when dealing with a single semiconductor supplier, different groups within the company may develop the libraries and they may not be consistent with each other. Logical libraries are often expressed in a format such as Synopsys library .db (or the ASCII .lib format) and these may not match the physical libraries in formats such as LEF, or GDSII, even in basic factors such as the number of pins. Cell timing models come in a variety of formats, i.e. Synopsys NL delay models, DCL from IBM, scaled polynomial models, and others. These models too must be accurately linked to the design at run time.

- Since the physical synthesis covers a wide range of design activities (RTL to GDSII) that were realized by various level of design precision, the library needs to provide consistent data for these algorithms. It is well known that physical synthesis does not eliminate the iteration of design cycle, it simply absorbs it inside the tool session. For example, in the early stage of logical optimization, the tool might use the statistical wire-load model to predict the loading, Once the floor planning or placement information is available, a better loading model can be derived based on the physical topology. These models are all required for physical synthesis.

- In some methodologies, the line between “library data” and “design data” gets blurry. For example, in a modern floorplanner, some blocks may be hierarchical, some hard IP from a library, and some may be “black boxes” which are designs that have some externally visible properties, but whose insides are unknown (because they are not yet synthesized) or unknowable (because they are proprietary IP).

- There are many other practical considerations: *E.g.* In commercial EDA systems, it is generally not OK just to detect mismatches and abort. Users demand friendly software that will do reasonable operation in the face of inconsistent data; Library systems are often used as repositories for design support information, such as company standards for colors, preferred direction of metal layers, etc. A “read only” library system may not be adequate for this.; EDA users demand efficient access to both the design and library data, so that they can solve their own problems and include their own proprietary enhancements without involving the EDA vendor.

The talk will look at the data model aspects of current and future systems for physical synthesis, including a tool that does timing aware floorplanning and a tool that converts RTL to placed gates. Examples of linking disjoint libraries with design databases in a realistic flow will be used to illustrate the challenges involved.