A unified data model for EDA tool integration

PATRICK GROENEVELD
MAGMA DESIGN AUTOMATION
PATRICK@MAGMA-DA.COM

ABSTRACT

Magma flagship RTL-to-GDS2 tool suite is built on an integrated data model. In a nutshell, this means that the programs ***share*** a single data structure that holds all design data such as cells, nets and wires. This is a major contrast to conventional databases that mainly serve as an interface format for bolting independent tools together. The unified data model approach not only avoids slow data format translations during the design flow execution, it also significantly reduces the programming effort for building the tool suite. The flexibility of the data model was the key enabler for the successful application of the Magma tools in the market.

The presentation will describe the architecture and basic concepts of the Magma unified data model. The major engineering choices made include the simplicity and the robustness. Both objectives were in part achieved using a strict object-oriented architecture. All physical objects (such as wire segments and cells) share a common base class that describes a rectangle. Large collections of cells can be efficiently searched using a KD-tree class data structure. The KD-tree area query serves as basis for GUI window redrawing, but is also used in the routers and the parasitic extractor.

All user interaction is performed using uniform TCL commands. This interface allows user access to all design data. Therefore it is powerful enough to program arbitrary EDA algorithms in TCL, such as ECO routing and spare cell placement. Also the GUI commands are interfaced through easy TCL strings.

Robustness of the data model requires special attention since commands can modify the data model in various ways. Usage through the TCL interface my never cause a crash. Another concern is the disk image of the data model. During normal operation, all design data is in core. Using a single command the entire data model is stored in a set of disk files called a *volcano*. Since the volcano contains all data and state information, execution can simply be resumed after loading the volcano. The internal format is binary, but special lisp-style structures enable a large degree of forward and backward compatibility.

The presentation will discuss the abovementioned issues in more detail. The practical experiences, the engineering trade-offs, as well as the practical experience will be elaborated.