The *POWER* of the Xilinx Foundation Series Software

Why is it so important to have fully integrated development tools?

by Karen Fidelak, Integrated Solutions Technical Marketing, Xilinx, karen.fidelak@xilinx.com and Loren Lacy, Integrated Solutions, Product Marketing, Xilinx, loren.lacy@xilinx.com

How would you define the ultimate set of design automation tools? You would probably include best-in-class point tools, the ability to run on a variety of operating systems and computing platforms, the ability to share data between operating systems and computing platforms, and the ability to seamlessly pass design data from one tool to the next.

Until recently, most companies focused more on the individual point tools than the flow of data between the tools. Large digital design companies would then connect each of the point tools in a customized flow. This need for connecting the various point tools led to the development and support of standard information exchange interfaces such as EDIF (Electronic Design Interchange Format). While EDIF is a very flexible standard its flexibility takes away from its simplicity when interfacing to various tools. So, while EDIF standardizes tool integration, it could be argued that it does not simplify the overall job.

Looking at the EDA landscape today, we see the beginning of a new focus. As more and more companies look at solving the "system on a chip" problem, it has become obvious that there is actually more value in the integration of tools than in the individual point tools themselves. This is apparent in the level of cooperation between traditional rivals such as Cadence and Synopsys, along with the demise of individual point tool providers like Zycad.

Design Flow Management

In addition to the easy passing of design information between point tools, you also value the ability to specify common information, just once, for multiple tools. This includes the location of simulation libraries, macro libraries, and timing information, for example. While the specification of this information can often be automated, updating one of the point tools within the flow often calls for a complete rewrite of the setup information. An integrated tool suite automatically communicates this information to each tool.

Using various tools within a design flow often requires the creation of additional design data files. Unlike a homegrown flow automation process, an integrated design suite is aware of the downstream tool requirements. For example, if you want to perform timing simulation after place and route, the place and route tools must be instructed to produce the timing simulation netlist, so it can be read by the simulator.

Project Management

Given the large number of source files, control files, and implementation files associated with a given design project, it becomes desirable and necessary to have the software tool take control of managing all of these various files.

A design project may consist of schematics, HDL files, IP cores, netlists, user constraints, or any combination of these. With the complexity of designs today, it can become difficult to manage all of these design modules as pieces of the design are modified. An integrated tool suite will know about all modules in the design and will determine when sources have been changed and therefore when design netlists must be updated and processes re-run. The tool will also clearly display all the design sources and implementation results and provide easy access to the appropriate editing tool for the given source file.

Schematic Entry tools as well as many HDL compilers require that the device family library be specified up front o provide the appropriate library symbols and components for the given architecture. Additionally, if a design is retargeted to a new device architecture in the middle of the design process, it is necessary to change the project libraries to match the new architecture. In an integrated tool suite this is done automatically by the tools, leaving you with nothing to do but select the device family, once. This selection will set the appropriate device libraries for design entry, as well as pass the device information forward to the place and route tools.

Designs are likely to be implemented many times in the course of a design cycle. For example, modifications to timing constraints, target device, and place and route options may be made in the pursuit of the best overall design implementation. As these design iterations are made, it is convenient to have access to previous results for comparison and archiving. An integrated tool suite provides revision control by archiving each implementation for future reference or use.

Optimization of the Design

You usually have some overall design strategy that you are looking to accomplish in your design flow. This strategy may place the highest priority on fitting the design in the smallest possible device, or getting the fastest performance, for example. The synthesis tool may be able to optimize the design based on the given timing requirements, but then the place and route tools also need to receive the same direction to appropriately place and route the design. This can mean setting these requirements twice. In an Integrated design environment, these settings need only be entered once.

Software Development

When placed in an integrated environment, software tools are given the difficult task of working well together, a task often left to the user with a collection of point tools. The tools must communicate with each other to pass design data. The data which is being communicated must be understood by all tools involved, and consideration must be taken for enabling the features and benefits of all tools involved. When an integrated tool suite is created, the software developers must collaborate. Interaction in the development phase is critical to ensure a working environment in the final product.

Software quality assurance is a collaborative effort in an integrated tool suite. If various partners' point tools are involved, the majority of the individual point tool testing is done by the respective vendor. Integration testing is done by all partners involved to ensure the proper flow of data between tools. An integrated tool suite undergoes thorough testing from design entry through design implementation to ensure that all tools communicate properly with each other and that data is successfully processed.

Software Support

Running into a problem with the software while doing a programmable logic design can be very frustrating. What can be even more frustrating is hunting around trying to find the vendor who can actually answer the question. With a true integrated tool environment, each vendor involved should be familiar with the entire design flow and with the other tools involved. All of the partners work very closely to develop and test the product. Support for the entire product should be available from a single source, where the support engineers are well trained in the individual point tools as well as the interfaces between them.

Xilinx Foundation Series – a Completely Integrated Tool Suite

The Xilinx Foundation Series provides software for every phase of the design flow including design entry (schematic, HDL, state machine), synthesis, simulation, implementation, and verification—all under a single project management environment. Synopsys FPGA Express HDL Synthesis and Aldec schematic entry and simulation are combined with Xilinx implementation tools to provide a first-class integrated design environment.

Libraries necessary for the selected device architecture are automatically included in the project, and synthesis optimizations are tailored to the selected device as well as overall design strategy. Timing constraint information, which is entered in the synthesis phase of the flow is passed directly, to the place and route tools. Files necessary for timing simulation or floorplanning are automatically created by the implementation tools and later loaded into the respective point tool upon invocation.

Source files and implementation revisions are marked in or out of date as appropriate, and automatic updating of the design is available when necessary.

These are just a few examples of features within the Foundation Series software which demonstrate the true integration of the product.

Conclusion

Software engineers from Synopsys, Aldec, and Xilinx work closely through the product development to ensure tight integration, ease of use, and high overall quality of results. The entire Foundation series design flow is fully tested by Xilinx SQA engineers, and support for all aspects of the flow is available from Xilinx technical support. The effort put forth by all partners involved has led to a tightly integrated, full-featured, well-supported product which ultimately increases the productivity of the end designer.