Accelerating RTL Reuse

Major Benefits:
- Practical RTL reuse through rapid understanding and easy deployment of existing RTL code:
  - Analyze design integrity
  - Assess RTL code quality
  - Visualize behavior and structure
- Automatic generation of graphical, HTML documentation for design reviews, archiving and reuse
- Flexible integration into any design flow via scripts or the powerful user interface
- Part of a complete solution for creating and managing complex Verilog, VHDL and mixed-language ASIC/FPGA designs

Challenges of Design Reuse

IP Reuse was heralded as a methodology to enable companies to vastly improve time to market and allow engineers the ability to adapt code to their needs by simply changing a few parameters. The simple fact is that design engineers do not have the time to write code specifically to enable future reuse. As a result, design teams turn to reusing or recycling existing RTL code.

Industry studies show that up to 80% of new ASIC/FPGA designs reuse code from a previous design. Despite this high percentage of RTL reuse within a company, engineering teams rarely spend the time necessary to insure sufficient design code quality or develop the necessary documentation, resulting in RTL code that is little more than a collection of files plus the recollection of a few key engineers. Consequently, valuable engineering time is spent analyzing the code's functionality just to understand its completeness and applicability to the current application — all towards deciding whether it is better to adapt the existing code or rewrite from a clean sheet. What is needed is a methodology that will accelerate and automate the process of evaluating and understanding RTL slated for reuse.

Enter HDL Designer

HDL Designer™ from Mentor Graphics is a comprehensive Verilog and VHDL design environment that enhances the productivity of engineering teams working on complex ASIC and FPGA designs. HDL Designer was developed to aid engineering teams in making the critical decision: to reuse or not to reuse. If the decision is to reuse code, then the HDL Designer environment can quickly identify any problematic areas as well as aid engineers in understanding and adapting the code to their specific application.

To reuse or not to Reuse

The process of automatically evaluating and understanding RTL code under consideration for reuse consists of three primary steps: Design Integrity Analysis, Code Quality Assessment and Design Visualization. At the completion of this process, a final step of Share Knowledge and Document is used to pass along the information resulting from the reuse analysis.
to help the next designer interested in utilizing the code in their design. Adopting this three-step methodology as part of a company’s design process leads to better RTL-reuse practices and productivity gains.

**Design Integrity Analysis**

An engineer, when tasked by management to evaluate existing code for reuse, may receive little more than a collection of files — no documentation, code not written to any standard, unknown design hierarchy and possibly missing files. Rather than being forced into the difficult and lengthy process of trying to compile the design into a simulator just to determine if there are syntax errors, missing libraries and files, HDL Designer will, in minutes, automatically comprehend the design hierarchy, highlight syntax errors, and point out missing or orphaned components. Armed with this information, an engineer can quickly determine code completeness as well as understand where the deficiencies are from an integrity point of view.

**Code Quality Assessment**

Instead of relying on a subjective analysis of how good the code under consideration is, HDL Designer analyzes code based on a selected design rule set. A standard part of HDL Designer is the inclusion of the Reuse Methodology Manual (RMM rev. 3.0), Xilinx and Altera design rule sets. In addition, HDL Designer supports customizable rule sets, allowing easy modification to fit a company's standards in RTL coding.

Detailed scoring metrics associated with the design rules provide an overall quality score, enabling engineers to identify potential problem areas as well as the ability to provide management with an objective, detailed analysis — enabling fact-based decisions on reuse.

**Design Visualization**

Once the decision has been made to adapt existing code to a new design, the design team now needs to analyze the code to understand its structure and behavior.

Visualization is key to understanding, and HDL Designer can quickly create different visualizations, depending upon the topology of the design and user preferences: hierarchy can be displayed as block diagrams, state machines as bubble charts, procedural code as flow charts, or the entire design can be viewed in an interface-based-design (IBD) spreadsheet. As the code is modified, these views are dynamically updated and can be easily exported to form a part of the design documentation package.

**Share Knowledge and Document**

Crucial to the design process is documentation for design reviews, communication with other design groups, as well as archiving for future reuse. HDL Designer automates a traditionally cumbersome task, allowing engineers to spend their time designing. Within minutes, a designer can create an interactive website describing their design both textually and graphically, enabling easy communication with managers, other designers and design teams.

**Accelerate RTL Reuse**

HDL Designer can support any design flow through HDL Designers' intuitive user interface or through scripting, enabling design teams to implement a practical design reuse strategy and avoiding costly impacts to both project schedules and budgets.