ABSTRACT

Designing Reusable VHDL Template Components

Onoriode Ikede & Kenneth B. Kent

In today's business world companies are under relentless pressure to deliver better and less expensive products and services to the marketplace faster. Introducing a product early to market makes a big difference in the product's profitability, since market windows for products are becoming increasingly short. For IT companies this means shortening the time spent in product development and increasing the productivity of the engineers. In software or hardware development the development time can be decreased by reusing existing Intellectual Property (IP). Reusing an existing IP can leads to an improved productivity (system not written from scratch), reliability (reusing parts that have been tested rigorously and verified) and shorten time to market (increased productivity leads to faster time to market).

The idea of reusing existing IP is not a new one in hardware development. Many hardware designers have successfully applied opportunistic reuse; reusing portions of their layouts, net lists or hardware description language code without significant modification. Opportunistic reuse works fine in a limited domain, but it does not scale as the domain expands. A truly reusable IP is a generic IP that can scale with its domain. The IP is reused with minimal modification to the original. This research focuses on adding a template capability to the VHDL language. The template VHDL component can therefore be used to scale and modify hardware components based on the parameter arguments of the template. The hardware developed using the template is a truly generic reusable IP because, the original hardware design specification is not modify to scale or reconfigure the hardware.