Contribution ID: **75** Type: **Oral**

Incremental Firmware Development and Partial Reconfiguration in the Xilinx Virtex-5

Friday 19 September 2008 09:25 (25 minutes)

The size and complexity of the latest generations of FPGAs has increased dramatically. This in turn means that the time taken to develop and build even small firmware projects is increasing exponentially. Pre-constrained logic placement and routing is becoming critically important for the use of specialized components in the FPGA such as serial link interfaces. This necessitates significant changes from 'normal' firmware tool flows in order to effectively develop systems based on these devices. In this paper we discuss several methods for improving turnaround speed and design safety, including: pre-placed and pre-routed hard macros / RPMs, pre-synthesised black-box netlists, and incremental synthesis, place and route. Possible methods of dynamic partial reconfiguration are also discussed in this context.

Summary

The current CMS trigger and DAQ electronics is based on FPGA and processor technology several generations behind the currently commercially-available devices. Most systems in CMS are based on either the Xilinx Virtex-II or the Xilinx Virtex-II Pro (and similar Altera-made FPGAs). Since then Xilinx has developed the Virtex-4 and Virtex-5, scaling in feature size from 250nm to 65nm and incorporating many new hardware features such as Digital Signal Processing (DSP) cores, tri-mode Gigabit Ethernet (GbE) MACs and PCI express (PCIe) endpoints amongst others. Furthermore, total logic capacity has increased approximately ten-fold with a corresponding increase in maximum clock speed.

However this creates a new problem: FPGA firmware synthesis and routing is an extremely computationally-intensive process. Given the arrangement of logic and routing in the devices and the scaling of feature size in deep submicron technology, the complexity of logic placement and routing is increasing at a greater-than-geometric rate. Without the development of new firmware development techniques to manage this increase in complexity, firmware build times can increase from minutes to hours and even days.

In order to avoid these problems we have been investigating the incorporation of pre-synthesised and pre-placed/routed modules (known as 'black box' modules, 'hard macros' or 'relationally-placed macros', depending on their construction) into a larger design. In addition we discuss the use of incremental synthesis, placement and routing, which rely on the use of a previously build design as a 'guide' for the next build. The benefits and difficulties of these various approaches are discussed.

Finally we discuss the application of these approaches in the concept of dynamic partial reconfiguration, which relies on the use of a 'static' module within a larger variable design, which allows internal reconfiguration of a device without disrupting the state of the module being used to provide general configuration and communication to the FPGA.

Author: Dr JONES, John (Princeton University)

Co-author: Mr STETTLER, Matthew (Los Alamos National Laboratory)

Presenter: Dr JONES, John (Princeton University)

Session Classification: Parallel Session B6 - Programmable Logic, boards, crates and systems