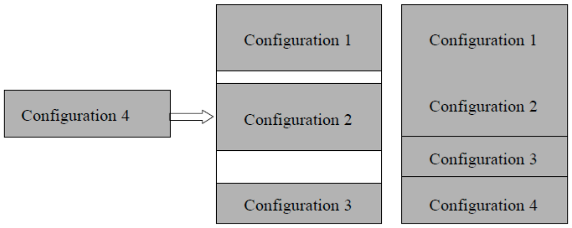
Configuration Caching Management Techniques for Reconfigurable Computing

Zhiyuan Li, Katherine Compton, Scott Hauck



Year of publication: *2000*
Area: *Run-Time Systems and Run-Time Configuration*

Anyone who has ever been on a program committee knows that there is one debate that almost always occurs, and that debate concerns the relative value of novelty and thoroughness. Most clever innovative papers are experimentally weak. The papers that have flawless experimental sections are usually incremental. I was not at the FCCM meeting in 2000, but this paper would not have created that discussion. It is both novel and thorough.

The idea is simple. Configuration time is so large that run time reconfiguration might not be able to effectively improve the performance of reconfigurable computing systems. In order to minimize the impact of configuration, this paper proposes and examines the possibility of configuration caching. To evaluate this, the authors have to construct a model for three different kinds of FPGA configuration structures: single context FPGAs, run-time reconfigurable FPGAs, and multi-context FPGAs. This was non-trivial because these models did not all exist commercially, and those that did exist were not comparable to each other. In addition, the paper evaluates both off-line and on-line optimizations of the configuration placement. Finally, the authors recognized the issue of configuration fragmentation, as illustrated above, and its impact on the ability to reconfigure. The paper is an exemplar of thorough evaluation.

The influence of this paper is an interesting subject. Twelve years after publication, there are no commercial systems that utilize cached configurations, at least not in the manner envisioned in this paper. However, neither are there commonly accepted run-time reconfiguration execution models that could benefit from these solutions. The multi-context FPGA that does exist is based on a different computational model. This paper is highly referenced by others who have attempted variations of the techniques presented here with different applications and algorithms. The less acknowledged value of research is to explore ideas that do not seem to pay off. Then again, perhaps this paper is still just ahead of its time. If we do eventually define suitable, standard execution models, this work provides a good first set of hints about architectures and algorithms to support them.

Herman Schmit

DOI: http://dx.doi.org/10.1109/FPGA.2000.903390